Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1982

A user's manual for interactive linear control programs on IBM/3033.

Thompson, Robert,mayor.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/20394

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

LIBRARY, NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A USER'S MANUAL FOR INTERACTIVE LINEAR CONTROL PROGRAMS ON IBM/3033

by

Robert M. Thompson

December 1982

Thesis Advisor: H. A. Titus

Approved for public release, distribution unlimited

T2C8809

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A User's Manual for Interactive Linear Control Programs on IBM/3033		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert M. Thompson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1982
		13. NUMBER OF PAGES 116
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Interactive; Kalman Filter; Luenberger Observer		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) There existed a need for an interactive program that would provide the user assistance in solving applications of linear control theory. The Linear Control Program (LINCON) and its user's guide satisfy this need. A series of ten interactive programs are presented which permit the user to carry out analysis, design and simulation of a broad class of linear control problems.		

LINCON consists of two groups: matrix manipulation, transfer function and time response programs; and modern controls programs. Examples for each are worked within each terminal session section.

Approved for Public Release, Distribution Unlimited.

A User's Manual for Interactive Linear Control Programs on IBM/3033

by

Robert M. Thompson
Lieutenant, United States Navy
B.S., University of Kentucky, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1982

LIBRARY, NAVAL POSTGRADUATE SCH
MONTEREY, CA 93940

ABSTRACT

There existed a need for an interactive program that would provide the user assistance in solving applications of linear control theory. The Linear Control Program (LINCON) and its user's guide satisfy this need. A series of ten interactive programs are presented which permit the user to carry out analysis, design and simulation of a broad class of linear control problems.

LINCON consists of two groups: matrix manipulation, transfer function and time response programs; and modern controls programs. Examples for each are worked within each terminal session section.

TABLE OF CONTENTS

I. INTRODUCTION 9

II. SYSTEM OVERVIEW 11

A. INTERACTIVE OPERATION 12

B. HIGH LEVEL LANGUAGE 12

C. MODULAR SOFTWARE 13

D. USER-ORIENTED OPERATION 14

E. INPUT RESTRICTIONS AND LIMITATIONS 14

III. MATRIX MANIPULATION, TRANSFER FUNCTION AND TIME RESPONSE PROGRAMS 17

A. INTRODUCTION 17

B. BASIC MATRIX PROGRAM (BASMAT) 19

1. Terminal Session Example 19

C. PARTIAL FRACTION EXPANSION PROGRAM (PRFEXP) . 24

1. Terminal Session Example 25

D. POLYNOMIAL ROOTS PROGRAM (ROOTS) 28

1. Terminal Session Example 28

E. RATIONAL TIME RESPONSE PROGRAM (RTRESP) . . . 30

1. Input Restrictions And Limitations 30

2. Terminal Session Example 31

IV. MODERN CONTROL PROGRAMS 38

A. INTRODUCTION 38

B. OBSERVABILTIY INDEX AND CONTROLLABILITY PROGRAM (OBSCON) 39
1. Input Restrictions 40
2. Terminal Session Example 40
C. OPTIMAL CONTROL/KALMAN FILTER PROGRAM (RICATI) 44
1. Terminal Session Example 46
D. DISCRETE TIME KALMAN FILTER (KALMAN) 53
1. Input Requirements 55
2. Terminal Session Example 56
E. STATE VARIABLE FEEDBACK PROGRAM (STVAR) . . . 61
1. Terminal Session Example 64
F. LUENBERGER OBSERVER PROGRAM (LUEN) 70
1. Terminal Session Example 73
G. OPTIMAL CONTROL PROGRAM (OPTCON) 85
1. Terminal Session Example 86
V. CONCLUSIONS AND RECOMMENDATIONS 92
A. CONCLUSIONS 92
B. RECOMMENDATIONS 93
APPENDIX A 95
APPENDIX B 96
APPENDIX C 97
APPENDIX D 98

APPENDIX E 100

APPENDIX F 101

APPENDIX G 104

APPENDIX H 106

APPENDIX I 108

APPENDIX J 111

LIST OF REFERENCES 114

BIBLIOGRAPHY 115

INITIAL DISTRIBUTION LIST 116

ACKNOWLEDGEMENT

I wish to express my sincere appreciation to Professor Hal Titus for his guidance and assistance during the preparation of this thesis.

I wish to dedicate this thesis to my wife, Sandi, and to our children, Treva, Melissa, and Robyn, for without their constant love, support, and devotion, this work would not have been possible.

I. INTRODUCTION

The purpose of this thesis was to update an existing program which provides assistance in solving computational problems associated with the study and application of linear control theory. The Linear Controls Program (LINCON) was first developed by Melsa [1] and adapted for batch use at NPS by Desjardins [2]. Although the original intent of this thesis was simply to take Desjardins' adapted version of Melsa's LINCON and further adapt it by making it interactive, LINCON soon began to grow as other routines were incorporated, as will be noted, until its present form was achieved.

LINCON, as such, is a high level applications software system made up of a large number of program tools for interactive analysis, design and simulation of a broad class of linear control problems. With LINCON, users can concentrate on their specialized applications rather than system design and routine program development, thereby saving valuable time.

It is assumed that the reader is familiar with the basic concepts of linear control theory as may be obtained from any one of a number of available textbooks (see the

bibliography). As such, the LINCON subprograms are presented in a user-oriented fashion. First, their purpose and some general rules that may apply are given; then the input requirements are presented and the expected outputs are described. Examples for each are worked out including a copy of the terminal session and the final results.

II. SYSTEM OVERVIEW

During LINCON's latest development, underlying guidelines called for concepts which accounted for the needs of the computer system, the programmer, and the user.

The guidelines followed during this latest development stage were:

(1) Operation of the system should be in an on-line interactive mode such that data can be easily input to (or output from) the system and readily accessed for verification, examination, and processing.

(2) Program development should be in a high-level language in order to facilitate software implementation and promote machine independence.

(3) The software should be modular in structure so that programs can be modified or inserted without affecting existing programs.

(4) Programs should be invoked by means of logical procedures or commands which minimize interaction time and which are user-oriented so that people can operate the system without first becoming computer experts.

A. INTERACTIVE OPERATION

The applications of LINCON are focused on interactive processing. Experience has shown that interactive on-line communication has many advantages in a research environment because it offers the opportunity to make observations and select alternate courses of action in a more flexible manner than with batch processing [3]. LINCON is organized around a collection of inter-related command programs, each of which performs a specified function and can be executed by means of a simple keyboard initiation sequence.

B. HIGH LEVEL LANGUAGE

An important feature in the design of LINCON is that it was implemented in a high level language. Program development in assembly language is more time consuming and results in system dependent software.

LINCON is programmed in ANSI standard FORTRAN and follows the conventions of FORTRAN IV. FORTRAN has been found to be a useful language for several reasons:

(1) Since some form of FORTRAN is available on most computers, LINCON is highly portable from one computer to another. A FORTRAN based system is helpful for importing programs as well as exporting

them. Of course, FORTRAN compilers don't all follow the same standards so there can still be difficulties.

(2) FORTRAN is a simple enough language that relatively complex programs can be implemented in a short period of time. Most scientific and research personnel know FORTRAN sufficiently well to write their own programs if necessary.

(3) Algorithms can be tested and implemented in FORTRAN and later converted to assembly language versions if more speed and efficiency are necessary. This procedure has the further benefit of aiding portability such that even if parts have been converted to assembler, equivalent FORTRAN versions are available.

C. MODULAR SOFTWARE

Overall system flexibilty is achieved by means of modularity. LINCON is actually made up of a large number of independent command programs. Each command program stands on its own with the ability to take some form of input, possibly supplied by some previous command, and generate some form of output, possibly to be used by a follow-up command.

D. USER-ORIENTED OPERATION

An important aspect in this modification of LINCON was to make the commands user-oriented so that operating the system does not require an engineering or computer background. This was achieved with a standard terminal keyboard by using phonetic characters which relate to the function which the command is to perform. A good combination of brevity and clarity is built into LINCON to avoid having to push extra buttons on the keyboard while at the same time preventing ambiguity.

E. INPUT RESTRICTIONS AND LIMITATIONS

Although the input requirements are fully described in the presentation of each program, there are several input format similarities used by all of them. For ease of use, and, more honestly, for ease of programming, most of the data input is grouped in the same arrangement.

The first input of every program is used to identify the problem for reference and for output data. A maximum of twenty alpha-numeric characters can be used. This restriction was not a system limitation but a programmer decision.

The next input common to all programs is the dimension of the plant matrix or A matrix. The format is I1 which would normally restrict the user to a maximum matrix size of 9x9, however, due to, again, a programmer decision, a dimension size not to exceed 8 is requested. The reasoning behind this was due, in part, to the printer. The NPS printer is capable of printing 133 characters on a line. Since the output format to the printer is 8E16.6 this naturally limits one to 8 numbers per line. Six places are normally considered necessary for good accuracy. A solution that would have lead to an unlimited matrix size would have been to incorporate a "wrap around" routine within the program. After attempting this it was decided the results were just too difficult to read.

Matrices are entered one element at a time beginning with element 1,1 and continuing across the row. The next row is then entered, and the process continues until all elements have been entered. After the matrix is entered, the complete matrix is automatically brought to the screen for review and possible correction. If a change to the matrix is desired the user simply enters the row number and column number without a separating comma. For example, 35 would

indicate the element in row 3 and column 5. After being prompted the change is entered. A review of the matrix is again screened. The user is again prompted for any possible changes. This procedure continues until all changes have been made.

Any special requirements or limitations will be brought to the user's attention within each program presentation.

III. MATRIX MANIPULATION, TRANSFER FUNCTION AND TIME RESPONSE PROGRAMS

A. INTRODUCTION

In this chapter four programs are discussed which may be used for the analysis and design of linear control systems represented in state variable form as

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{b}u(t) \tag{3.A-1}$$

$$u(t) = K[r(t) - \underline{k}^T\underline{x}(t)] \tag{3.A-2}$$

$$y(t) = \underline{c}^T\underline{x}(t) \tag{3.A-3}$$

The first, BASMAT, is the Basic Matrix manipulation program which is used for computing the determinant, inverse, characteristic polynomial, and eigenvalues for a square matrix $\underline{A}$. In addition, BASMAT will calculate the state transition matrix and the PHI(s) matrix. The second program, PRFEXP, calculates the partial fraction expansion of a polynomial. The third program, ROOTS, calculates the roots of a polynomial. The fourth program is used for determining the time response of linear control systems. RTRESP will determine the rational time response of a system in closed-form provided that the input function r(t) has a

rational time response and that there be no repeated eigenvalues in the combination of the system and input. It should be noted here, and will be again in the actual discussion of the program, that by setting r(t) and K equal to zero, unforced and open-loop systems may be studied, respectfully.

B. BASIC MATRIX PROGRAM (BASMAT)

Given the plant matrix $\underline{A}$, BASMAT can compute the following:

(1) the determinant of $\underline{A}$, det $\underline{A}$

(2) the inverse of $\underline{A}$, $\underline{A}^{-1}$

(3) the characteristic polynomial, det($s\underline{I}-\underline{A}$)

(4) the eigenvalues of the characteristic polynomial, λ

(5) the state transition matrix, $\underline{\Phi}(t) = \exp(\underline{A}t)$

(6) the PHI(s) matrix, $\underline{\bar{\Phi}}(s) = (s\underline{I}-\underline{A})^{-1}$

1. Terminal Session Example

This section contains a terminal session for a specific problem. Commands entered by the user are in lower case. All of BASMAT's capabilities will be utilized beginning with following plant matrix:

$$\underline{A} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & -2.0 & 1.0 \\ 0.0 & -0.5 & 1.0 \end{bmatrix}$$

```
lincon

EXECUTION BEGINS...
 LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
    BASIC MATRIX MANIPULATION - <BASMAT>
    RATIONAL TIME RESPONSE - <RTRESP>
    STATE VARIABLE FEEDBACK - <STVAR>
    CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
    LUENBERGER OBSERVER - <LUEN>
    OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
    DISCRETE TIME KALMAN FILTER - <KALMAN>
    OPTIMAL CONTROL - <OPTCON>
    PARTIAL FRACTION EXPANSION - <PRFEXP>
    ROOTS OF A POLYNOMIAL - <ROOTS>
 TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
```

```
SYMBOLS < >.
basmat

   BASMAT PROVIDES MATRIX MANIPULATION TO SOLVE
   FOR DETERMINANTS, INVERSES, STATE TRANSITION AND
   PHI(S) MATRICES, EIGENVALUES, AND CHARACTERISTIC
   POLYNOMIALS.

   FIRST ENTER THE PROBLEM IDENTIFICATION
   (*NOT TO EXCEED 20 CHARACTERS*)
thesis example

   NOW ENTER THE DIMENSION OF THE A MATRIX (UP TO 8).
3

   THE ELEMENT A(1,1) =
1.0

   THE ELEMENT A(1,2) =
0.0

   THE ELEMENT A(1,3) =
0.0

   THE ELEMENT A(2,1) =
0.0

   THE ELEMENT A(2,2) =
-2.0

   THE ELEMENT A(2,3) =
1.0

   THE ELEMENT A(3,1) =
0.0

   THE ELEMENT A(3,2) =
-0.5

   THE ELEMENT A(3,3) =
1.0




    THE A MATRIX
        1.00E+00   0.0         0.0
        0.0       -2.00E+00   1.00E+00
        0.0       -5.00E-01   1.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   DO YOU WANT TO CALCULATE THE DETERMINANT?
y

    THE DETERMINANT OF THE MATRIX
      -1.50E+00

   ARE YOU SATISFIED WITH THE RESULTS?
y

   DO YOU WANT THE DETERMINANT PRINTED?
y

   DO YOU WANT TO CALCULATE THE INVERSE?
y
```

```
     THE INVERSE OF THE MATRIX

         1.00E+00  0.0          0.0
         0.0      -6.67E-01  6.67E-01
         0.0      -3.33E-01  1.33E+00

    ARE YOU SATISFIED WITH THE RESULTS?
 y
    DO YOU WANT THE INVERSE PRINTED?
 y
    DO YOU WANT TO CALCULATE THE PHI(S) MATRIX?
 y
*****************************************************************

     THE MATRIX COEFFICIENTS OF THE NUMERATOR OF THE
     PHI(S) MATRIX

     THE MATRIX COEFFICIENT OF S**2

         1.00E+00  0.0          0.0
         0.0       1.00E+00  0.0
         0.0       0.0          1.00E+00

     THE MATRIX COEFFICIENT OF S**1

         1.00E+00  0.0          0.0
         0.0      -2.00E+00  1.00E+00
         0.0      -5.00E-01  1.00E+00

     THE MATRIX COEFFICIENT OF S**0

        -1.50E+00  0.0          0.0
         0.0       1.00E+00 -1.00E+00
         0.0       5.00E-01 -2.00E+00

    ARE YOU SATISFIED WITH THE RESULTS?
    A NO RESPONSE WILL GIVE YOU THE OPTION TO MAKE
    CHANGES TO THE A MATRIX.
 y
    DO YOU WANT A PRINTOUT OF THE RESULTS?
 y
    DO YOU WANT TO CALCULATE THE CHARACTERISTIC POLYNOMIAL?
 y
*****************************************************************

     THE CHARACTERISTIC POLYNOMIAL-IN ASCENDING POWERS OF S

         1.50E+00 -2.50E+00  0.0        1.00E+00

    ARE YOU SATISFIED WITH THE RESULTS?
 y

    DO YOU WANT THE CHARACTERISTIC POLYNOMIAL PRINTED?
 y
    DO YOU WANT TO CALCULATE THE EIGENVALUES?
 y
*****************************************************************
```

```
     THE EIGENVALUES OF THE A MATRIX
        REAL PART IMAG. PART
         8.23E-01  0.0
        -1.82E+00  0.0
         1.00E+00  0.0

    ARE YOU SATISFIED WITH THE RESULTS?
 y
    DO YOU WANT THE EIGENVALUES PRINTED?
 y
    DO YOU WANT TO CALCULATE THE STATE TRANSITION MATRIX?
 y
************************************************************
     THE ELEMENTS OF THE STATE TRANSITION MATRIX

     THE MATRIX COEFFICIENT OF EXP(  8.23E-01)T

         0.0         0.0       0.0
         0.0       -6.59E-02  3.78E-01
         0.0       -1.39E-01  1.07E+00

     THE MATRIX COEFFICIENT OF EXP( -1.82E+00)T

         0.0         0.0       0.0
         0.0         1.07E+00 -3.78E-01
         0.0         1.39E-01 -6.59E-02

     THE MATRIX COEFFICIENT OF EXP(  1.00E+00)T

         1.00E+00  0.0         0.0
         0.0       -2.26E-06  2.36E-06
         0.0       -5.36E-07  1.07E-06

    ARE YOU SATISFIED WITH THE RESULTS?
    A NO RESPONSE WILL GIVE THE OPTION TO MAKE
    CHANGES TO THE A MATRIX.
 y
    DO YOU WANT A PRINTOUT OF THE RESULTS?
 y
    THIS CONCLUDES THE BASIC MATRIX MANIPULATION PROGRAM
    (BASMAT).

    ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
 n
    LINCON IS NOW TERMINATED.
```

The computer results are shown in Appendix A. Interpretation of the determinant, inverse and eigenvalues are straightforward. The PHI(s) matrix is a combination of the numerator coefficients and characteristic polynomial. The results can be interpreted as

$$\underline{\Phi}(s) = \frac{1}{s^2-2.5s+1.5} \begin{bmatrix} s^2+s-1.5 & 0.0 & 0.0 \\ 0.0 & s^2-2.0s+1 & s-1 \\ 0.0 & -0.5s+0.5 & s^2+s-2 \end{bmatrix}$$

and, simlarily, the fifth term of the state transition matrix is

$$\underline{\Phi}(t) = -0.067\exp(0.82t)+1.07\exp(-1.82t)-2.26\times10^{-6}\exp(t)$$

C. PARTIAL FRACTION EXPANSION PROGRAM (PRFEXP)

PRFEXP can calculate a partial fraction expansion given a rational ratio of two polynomials in the form

$$G(s) = K \frac{N(s)}{D(s)} \tag{3.C-1}$$

where K = the input function gain, a constant

$N(s) = n_1 + n_2 s + n_3 s^2 + \ldots + s^q$

and $D(s) = d_1 + d_2 s + d_3 s^2 + \ldots + s^p$

The numerator and denominator coefficients must be arranged so that the coefficients of s^q and s^p are each unity with $p > q \geq 0$. The outputs of the program are

(1) the numerator gain

(2) the numerator polynomial and its roots; roots are considered equal if their real and imaginary parts do not differ by more than 0.005.

(3) the denominator polynomial and its roots; multiple roots are listed once, along with its multiplicity

(4) the numerator coefficients (residue matrix) are listed in the same order as the denominator roots; the first coefficient in any row of the matrix is the first-order term coefficient.

1. Terminal Session Example

The partial fraction expansion of the following rational polynomial is to be performed:

$$G(s) = \frac{14(s^2+2)(s+1)}{2s^4+6s^3+2s+5}$$

Putting the polynomial in a usable form yields

$$G(s) = \frac{7(s^3+s^2+2s+2)}{s^4+3s^3+s+2.5}$$

```
lincon

EXECUTION BEGINS...
LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
   BASIC MATRIX MANIPULATION - <BASMAT>
   RATIONAL TIME RESPONSE - <RTRESP>
   STATE VARIABLE FEEDBACK - <STVAR>
   CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
   LUENBERGER OBSERVER - <LUEN>
   OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
   DISCRETE TIME KALMAN FILTER - <KALMAN>
   OPTIMAL CONTROL - <OPTCON>
   PARTIAL FRACTION EXPANSION - <PRFEXP>
   ROOTS OF A POLYNOMIAL - <ROOTS>
TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
SYMBOLS < >.
prfexp

   PRFEXP IS USED TO DETERMINE THE PARTIAL
   FRACTION EXPANSION OF THE RATIO OF TWO POLYNOMIALS.

   FIRST ENTER THE PROBLEM IDENTIFICATION
   (*NOT TO EXCEED 20 CHARACTERS*).
thesis example

   ENTER THE INPUT FUNCTION GAIN--K.
7.

    THE INPUT FUNCTION GAIN
        7.00E+00

   DO YOU WANT TO CHANGE THE VALUE OF THE GAIN?
n

   ENTER THE NUMERATOR BY POLYNOMIAL COEFFICIENT OR
   FACTORED ROOT FORM. FIRST ENTER EITHER THE LETTER P
   FOR POLYNOMIAL COEFFICIENT FORM  OR THE LETTER F FOR
   FACTORED ROOT FORM.
p
```

```
   ENTER THE NUMERATOR POLYNOMIAL ORDER.
3

   THE POLYNOMIAL COEFFICIENTS MUST BE ENTERED IN ASCENDING
   ORDER OF S.

   **WARNING--THE HIGHEST ORDER COEFFICIENT MUST BE UNITY.*
   DO YOU NEED TO CHANGE THE INPUT FUNCTION GAIN TO SATISFY
   THIS REQUIREMENT?
n

   ENTER THE POLYNOMIAL COEFFICIENTS IN ASCENDING
   ORDER OF S.
   COEFF(1)=
2.

   COEFF(2)=
2.

   COEFF(3)=
1.

   COEFF(4)=
1.

    NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
2.00E+00  2.00E+00  1.00E+00  1.00E+00


    NUMERATOR ROOTS ARE
       REAL PART IMAG. PART
        0.0       -1.41E+00
        0.0        1.41E+00
       -1.00E+00  0.0

   ENTER THE DENOMINATOR BY POLYNOMIAL COEFFICIENT OR
   FACTORED ROOT FORM. FIRST ENTER EITHER THE LETTER P FOR
   POLYNOMIAL COEFFICIENT FORM OR THE LETTER F FOR FACTORED
   ROOT FORM.
p

   ENTER THE DENOMINATOR POLYNOMIAL ORDER.
4

   THE POLYNOMIAL COEFFICIENTS MUST BE ENTERED IN ASCENDING
   ORDER OF S.

   **WARNING--THE HIGHEST ORDER COEFFICIENT MUST BE UNITY.*
   DO YOU NEED TO CHANGE THE INPUT FUNCTION GAIN TO SATISFY
   THIS REQUIREMENT?
n

   ENTER THE POLYNOMIAL COEFFICIENTS IN ASCENDING
   ORDER OF S.
   COEFF(1)=
2.5

   COEFF(2)=
1.

   COEFF(3)=
0

   COEFF(4)=
3.

   COEFF(5)=
1.
```

```
     DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 2.50E+00  1.00E+00  0.0       3.00E+00  1.00E+00

     DENOMINATOR ROOTS ARE
        REAL PART IMAG. PART  MULTIPLICITY
       -3.019E+00 0.0             1
       -9.128E-01 0.0             1
        4.658E-01-8.308E-01       1
        4.658E-01 8.308E-01       1

     RESIDUE MATRIX - REAL PART
         5.81E+00
         3.17E-01
         4.36E-01
         4.36E-01

     RESIDUE MATRIX - IMAG. PART
        -2.05E-07
         0.0
         2.06E+00
        -2.06E+00

    THIS CONCLUDES THE PARTIAL FRACTION EXPANSION PROGRAM
    (PRFEXP).

    DO YOU WANT TO RUN THE PROGRAM AGAIN?
 n

    ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
 n

    LINCON IS NOW TERMINATED.
```

The results are shown in Appendix B. Interpretation of these results are:

$$G(s) = \frac{s-5.81+j0}{s+3.02} + \frac{s-0.32}{s+9.13} + \frac{s-0.44-j2.06}{s-0.47+j0.83} + \frac{s-0.44+j2.06}{s-0.47-j0.83}$$

D. POLYNOMIAL ROOTS PROGRAM (ROOTS)

ROOTS can calculate the roots of a polynomial of degree less than or equal to eight. Given a polynomial in the form

$$P(s) = p_1 + p_2 s + p_3 s^2 + \ldots + s^q \qquad (3.D-1)$$

the coefficient of s^q must be unity. The program output lists the polynomial coefficients for reference and the real and imaginary roots.

1. Terminal Session Example

The following polynomial is to be factored:

$$P(s) = -5s^3 + 15s^2 - 25.5s + 4$$

Putting the polynomial in the required form yields

$$P(s) = s^3 - 3s^2 + 5.1s - 0.8$$

```
lincon

EXECUTION BEGINS...
LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
   BASIC MATRIX MANIPULATION - <BASMAT>
   RATIONAL TIME RESPONSE - <RTRESP>
   STATE VARIABLE FEEDBACK - <STVAR>
   CONTROLIABILITY AND OBSERVABILITY - <OBSCON>
   LUENBERGER OBSERVER - <LUEN>
   OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
   DISCRETE TIME KALMAN FILTER - <KALMAN>
   OPTIMAL CONTROL - <OPTCON>
   PARTIAL FRACTION EXPANSION - <PRFEXP>
   ROOTS OF A POLYNOMIAL - <ROOTS>
TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
SYMBOLS < >.
roots
```

```
   ROOTS IS USED TO FIND THE ROOTS OF A POLYNOMIAL OF
   DEGREE LESS THAN OR EQUAL TO EIGHT.

   FIRST ENTER THE PROBLEM IDENTIFICATION
   (NOT TO EXCEED 20 CHARACTERS).
thesis example

   ENTER THE ORDER OF THE SYSTEM (UP TO 8).
3

   ROOTS OF A POLYNOMIAL
   ENTER THE POLYNOMIAL COEFFICIENTS IN ASCENDING
   ORDER OF S.

   **WARNING--THE HIGHEST ORDER COEFFICIENT MUST BE UNITY.*
   COEFF(1)=
.8

   COEFF(2)=
5.1

   COEFF(3)=
3.

   COEFF(4)=
1.

    POLYNOMIAL COEFFICIENTS - IN ASCENDING POWERS OF S
        8.00E-01  5.10E+00  3.00E+00  1.00E+00

    THE ROOTS ARE        REAL PART IMAG. PART
                         -1.41E+00  1.62E+00
                         -1.41E+00 -1.62E+00
                         -1.74E-01  0.0

   THIS CONCLUDES THE POLYNOMIAL ROOTS PROGRAM.

   DO YOU WANT TO RUN THE PROGRAM AGAIN?
n

   ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
n

   LINCON IS NOW TERMINATED.
```

The results shown in Appendix C are easily interpreted.

E. RATIONAL TIME RESPONSE PROGRAM (RTRESP)

The time response in closed form of a linear control system described by Eqs. (3.A-1), (3.A-2) and (3.A-3) is calculated by this program. The user must define the initial conditions $\underline{x}(0)$ and the rational Laplace transform of the scalar forcing function r(t). The theoretical concepts involved in the development of the computer codes are described by Melsa [1].

1. Input Restrictions And Limitations

Enter the following in F format (floating point):

1) the elements of the plant matrix $\underline{A}$
2) the control vector $\underline{b}$
3) the output vector $\underline{c}$
4) the feed backback coefficent vector $\underline{k}$
5) the controller gain K

and

6) the initial conditions vector $\underline{x}(0)$.

The rational Laplace transform of the input function must be in the form

$$\mathcal{L}\,[r(t)] = R(s) = G\,\frac{N(s)}{D(s)} \qquad (3.E-1)$$

where G = the input function gain, a constant

$N(s) = n_1 + n_2 s + n_3 s^2 + \ldots + s^q$

and $D(s) = d_1 + d_2 s + d_3 s^2 + \ldots + s^p$

The numerator and denominator coefficients must be arranged so that the coefficients of s^q and s^p are each unity with $p > q \geq 0$. Due to programming limitations it is necessary that the order of the system plus the dimension of D(s) be less than or equal to ten.

Upon entering the input function gain, the user next has the option to enter the numerator and denominator in either polynomial coefficient form or factored root form. With the factored root form enter the real part of the root as negative if it lies in the left half plane and just the magnitude of the imaginary part.

2. Terminal Session Example

This section contains a terminal session for a specific example. Commands entered by the user are in lower case. The RTRESP program will determine the time response form of the closed-loop system

$$\dot{\underline{x}}(t) = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & -2.0 & 1.0 \\ 0.0 & -0.5 & 1.0 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} u(t)$$

$$u(t) = 3.2\{r(t) - [1.0 \quad 1.0 \quad 0.0]\,\underline{x}(t)\}$$

$$\underline{y}(t) = [1.0 \quad 0.4 \quad 1.0]\,\underline{x}(t)$$

if the Laplace transform of the input function is

$$R(s) = 0.5 \; \frac{s + 1.0}{s^2 - 2.0}$$

and the initial conditions are zero.

Note here that the open-loop rational time response may be calculated by setting the feedback coefficient vector $\underline{k}$ equal to a zero vector. Also, if only an initial condition response is desired, the input function gain G is set to zero.

```
lincon

EXECUTION BEGINS...
 LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
    BASIC MATRIX MANIPULATION - <BASMAT>
    RATIONAL TIME RESPONSE - <RTRESP>
    STATE VARIABLE FEEDBACK - <STVAR>
    CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
    LUENBERGER OBSERVER - <LUEN>
    OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
    DISCRETE TIME KALMAN FILTER - <KALMAN>
    OPTIMAL CONTROL - <OPTCON>
    PARTIAL FRACTION EXPANSION - <PRFEXP>
    ROOTS OF A POLYNOMIAL - <ROOTS>
 TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
 SYMBOLS < >.
 rtresp
```

```
   RTRESP DETERMINES THE TIME RESPONSE OF A LINEAR
   FEEDBACK CONTROL SYSTEM. THIS PROGRAM WILL GIVE A
   CLOSED-FORM EXPRESSION FOR THE TIME RESPONSE.

   FIRST ENTER THE PROBLEM IDENTIFICATION
   (NOT TO EXCEED 20 CHARACTERS).
thesis example

   ENTER THE ORDER OF THE SYSTEM (UP TO 8).
3

   ENTER THE ELEMENTS OF THE PLANT MATRIX--A.

   A(1,1) =
1.0

   A(1,2) =
0.0

   A(1,3) =
0.0

   A(2,1) =
0.0

   A(2,2) =
-2.0

   A(2,3) =
1.0

   A(3,1) =
0.0

   A(3,2) =
-0.5

   A(3,3) =
1.0

    THE A MATRIX (PLANT MATRIX)

         1.00E+00  0.0          0.0
         0.0      -2.00E+00   1.00E+00
         0.0      -5.00E-01   1.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT(S) OF THE MATRIX?
n

   ENTER THE ELEMENTS OF THE CONTROL VECTOR--B.

   B(1) =
0.0

   B(2) =
0.0

   B(3) =
1.0

    THE B MATRIX (CONTROL VECTOR)

         0.0
         0.0
         1.00E+00


   DO YOU WANT TO CHANGE ANY ELEMENT(S) OF THE MATRIX?
n
```

```
   ENTER THE ELEMENTS OF THE OUTPUT VECTOR--C.
   C(1) =
1.0
   C(2) =
0.4
   C(3) =
1.0
    THE C MATRIX (OUTPUT VECTOR)

         1.00E+00
         4.00E-01
         1.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT(S) OF THE MATRIX?
n

   ENTER THE ELEMENTS OF THE FEEDBACK COEFFICIENT VECTOR --
   FDBG.
   FDBG(1) =
1.0
   FDBG(2) =
1.0
   FDBG(3) =
0.0
    THE FEEDBACK COEFFICIENT VECTOR

         1.00E+00
         1.00E+00
         0.0

   DO YOU WANT TO CHANGE ANY ELEMENT(S) OF THE VECTOR?
n
   ENTER THE CONTROLLER GAIN--K.
3.2
    THE CONTROLLER GAIN

         3.20E+00

   DO YOU WANT TO CHANGE THE VALUE OF THE GAIN?
n

   ENTER THE ELEMENTS OF THE INITIAL CONDITIONS VECTOR --
   X(0).
   XO(1) =
0.0
   XO(2) =
0.0
   XO(3) =
0.0
    INITIAL CONDITIONS VECTOR-X(0)

         0.0
         0.0
         0.0
```

```
   DO YOU WANT TO CHANGE ANY ELEMENT(S) OF THE VECTOR?
n


    ENTER THE GAIN FOR THE RATIONAL LAPLACE
    TRANSFORM OF THE INPUT FUNCTION.
0.5

   THE INPUT FUNCTION GAIN =
       5.00E-01

   DO YOU WANT TO CHANGE THE VALUE OF THE GAIN?
n


   ENTER THE NUMERATOR BY POLYNOMIAL
   COEFFICIENT OR FACTORED ROOT FORM.
   FIRST ENTER EITHER THE LETTER P FOR
   POLYNOMIAL COEFFICIENT FORM OR THE LETTER F FOR
   FACTORED ROOT FORM.
f

   ENTER THE NUMERATOR POLYNOMIAL ORDER.
1

   ENTER THE REAL PART OF THE ROOT.
-1.0

   ENTER THE MAGNITUDE OF THE IMAGINARY ROOT.
0.0

    NUMERATOR POLYNOMIAL OF R(S)-ASCENDING POWERS OF S

        1.00E+00  1.00E+00

    NUMERATOR ROOTS ARE
      REAL PART IMAG. PART
       -1.00E+00  0.0

   ENTER THE DENOMINATOR BY POLYNOMIAL
   COEFFICIENT OR FACTORED ROOT FORM.
   FIRST ENTER EITHER THE LETTER P FOR
   POLYNOMIAL COEFFICIENT FORM OR THE LETTER F FOR
   FACTORED ROOT FORM.
p

   ENTER THE DENOMINATOR POLYNOMIAL ORDER.
2

   THE POLYNOMIAL COEFFICIENTS MUST BE
   ENTERED IN ASCENDING ORDER OF S.

   WARNING--THE HIGHEST ORDER COEFFICIENT MUST BE UNITY.
   DO YOU NEED TO CHANGE THE INPUT
   FUNCTION GAIN TO SATISFY THIS REQUIREMENT?
n

   ENTER THE POLYNOMIAL COEFFICIENTS IN
   ASCENDING ORDER OF S.
   CO(1)=
-2.0

   CO(2)=
0.0

   CO(3)=
1.0
```

```
    DENOMINATOR POLYNOMIAL OF R(S) - ASCENDING POWERS OF S

       -2.00E+00  0.0        1.00E+00


    DENOMINATOR ROOTS ARE
      REAL PART IMAG. PART
        1.41E+00  0.0
       -1.41E+00  0.0
*****************************************************************
    THE TIME RESPONSE OF THE STATE X(T)

    THE VECTOR COEFFICIENT OF EXP(-5.0E-01)T*COS(1.2E+00)T

        0.0       -3.70E-01 -9.72E-01

    THE VECTOR COEFFICIENT OF EXP(-5.0E-01)T*SIN(1.2E+00)T

        0.0       -3.47E-01 -7.50E-02

    THE VECTOR COEFFICIENT OF EXP(  1.00E+00)T

        0.0        0.0        2.68E-06

    THE VECTOR COEFFICIENT OF EXP(  1.41E+00)T

        0.0        2.67E-01  9.12E-01

    THE VECTOR COEFFICIENT OF EXP( -1.41E+00)T

        0.0        1.03E-01  5.00E-02
*****************************************************************
    THE TIME RESPONSE OF THE OUTPUT Y(T)

    THE COEFFICIENT OF EXP( -5.00E-01)T*COS(  1.20E+00)T

       -1.12E+00

    THE COEFFICIENT OF EXP( -5.00E-01)T*SIN(  1.20E+00)T

       -2.14E-01

    THE COEFFICIENT OF EXP(  1.00E+00)T

        2.68E-06

    THE COEFFICIENT OF EXP(  1.41E+00)T

        1.02E+00

    THE COEFFICIENT OF EXP( -1.41E+00)T

        1.01E-01
*****************************************************************
   THIS CONCLUDES THE RATIONAL TIME RESPONSE PROGRAM
   (RTRESP)

   ANALYSIS IS COMPLETE.  DO YOU  WANT TO RUN LINCON AGAIN?
n

   LINCON IS NOW TERMINATED.
```

The computer results shown in Appendix D are interpreted as:

$$x_1(t) = 0.0$$

$$x_2(t) = -0.37\exp(-0.5t)\cos(1.2t) - 0.35\exp(-0.5t)\sin(1.2t) + 0.67\exp(1.41t) + 0.10\exp(-1.41t)$$

$$x_3(t) = -0.97\exp(-0.5t)\cos(1.2t) - 0.08\exp(-0.5t)\sin(1.2t) + 2.7\times10^{-6}\exp(t) + 0.91\exp(1.41t) + 0.06\exp(-1.41t)$$

$$y(t) = -1.12\exp(-0.5t)\cos(1.2t) - 0.21\exp(-0.5t)\sin(1.2t) + 2.7\times10^{-6}\exp(t) + 1.02\exp(1.41t) + 0.1\exp(-1.41t)$$

IV. MODERN CONTROL PROGRAMS

A. INTRODUCTION

In this chapter six programs are presented and discussed which may be used for the analysis and design of control systems.

The first, OBSCON, is used to find the observability index and controllability of a system. The next two programs are used to design Kalman filters. RICATI and KALMAN can find the feedback and control gains necessary to optimize a function for either continuous or discrete systems, respectively. The last three programs may be used to design optimal linear control systems. STVAR is particularly useful in the design of linear state variable feedback control systems. It may be used to calculate both open- and closed-loop transfer functions and also has the ability to design a closed-loop system from desired transfer function specifications. LUEN is used to design a combined, reduced-order, observer-controller to achieve a desired closed-loop transfer function from a system where some of the states are inaccessible. OPTCON will minimize a given cost function producing a scalar control.

B. OBSERVABILTIY INDEX AND CONTROLLABILITY PROGRAM (OBSCON)

This program is used to determine the observability index and controllability of the linear system

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{B}\underline{u}(t) \tag{4.B-1}$$

$$\underline{y}(t) = \underline{C}\underline{x}(t) \tag{4.B-2}$$

where

$\underline{x}$ = state vector (n-dimensional vector)

$\underline{u}$ = control vector (l-dimensional vector)

$\underline{y}$ = output vector (m-dimensional vector)

$\underline{A}$ = n x n matrix

$\underline{B}$ = n x l matrix

$\underline{C}$ = m x n matrix

The observability index is defined as the minimum integer such that the matrix

$$[\underline{C}, \underline{A}^T \underline{C}, \ldots, (\underline{A}^T)^{r-1}\underline{C}]$$

has rank n. The above system is said to be controllable at a given initial time if it is possible, by using an

unconstrained control vector, to force the system from an initial state of $\underline{x}(0)$ to some other state in a finite time interval [4].

If the user desires just the observability index to be calculated, enter the B matrix as a zero matrix. Likewise, if just the controllability of the system is desired, enter the C matrix as a zero matrix. The unobservable or uncontrollable system response is then of course ignored.

1. Input Restrictions

Other than the limitations to the problem identification, (20 characters) and system size (8x8) there are no restrictions to OBSCON. A reminder here may be helpful, however. Remember to enter your elements of the matrices in F format.

2. Terminal Session Example

This section contains a session for a specific problem. Commands entered by the user are in lower case. The following system is to be tested:

$$\underline{x}(t) = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ -1.0 & -0.5 & 1.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 2.0 & 1.0 \\ 0.0 & 1.0 \\ 0.0 & 0.0 \end{bmatrix} \underline{u}(t)$$

$$\underline{y}(t) = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & 1.0 & 0.0 \\ -2.0 & 1.0 & 0.0 \end{bmatrix} \underline{x}(t)$$

```
 lincon

 EXECUTION BEGINS...
  LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
    BASIC MATRIX MANIPULATION - <BASMAT>
    RATIONAL TIME RESPONSE - <RTRESP>
    STATE VARIABLE FEEDBACK - <STVAR>
    CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
    LUENBERGER OBSERVER - <LUEN>
    OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
    DISCRETE TIME KALMAN FILTER - <KALMAN>
    OPTIMAL CONTROL - <OPTCON>
    PARTIAL FRACTION EXPANSION - <PRFEXP>
    ROOTS OF A POLYNOMIAL - <ROOTS>
 TO USE  ONE OF THE SUBPROGRAMS  ENTER THE NAME  BETWEEN THE
SYMBOLS < >.
 obscon

    OBSCON DETERMINES THE OBSERVABILITY INDEX AND
    CONTROLLABILITY OF A SYSTEM.

    FIRST ENTER THE PROBLEM IDENTIFICATION
    (*NOT TO EXCEED 20 CHARACTERS*)
 thesis example

    NOW, ENTER THE ORDER OF THE SYSTEM (UP TO 8).
 3

    ENTER THE ELEMENTS OF THE PLANT MATRIX--A.

    A(1,1) =
 0.0

    A(1,2) =
 1.0

    A(1,3) =
 0.0

    A(2,1) =
 -1.0

    A(2,2) =
 -0.5

    A(2,3) =
 1.0

    A(3,1) =
 0.0

    A(3,2) =
 0.0

    A(3,3) =
 1.0
```

```
    THE A  MATRIX (PLANT MATRIX)
 0.0          1.00E+00   0.0
-1.00E+00  -5.00E-01   1.00E+00
 0.0          0.0        1.00E+00

   DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE NUMBER OF COLUMNS OF THE B MATRIX.
2

   ENTER THE ELEMENTS OF THE B MATRIX.
   B(1,1) =
2.0

   B(1,2) =
1.0

   B(2,1) =
0.0

   B(2,2) =
1.0

   B(3,1) =
0.0

   B(3,2) =
0.0

    THE B  MATRIX

 2.00E+00   1.00E+00
 0.0        1.00E+00
 0.0        0.0

   DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE NUMBER OF OUTPUTS.
3

   ENTER THE ELEMENTS OF THE C MATRIX.
   C(1,1) =
0.0

   C(1,2) =
1.0

   C(1,3) =
0.0

   C(2,1) =
1.0

   C(2,2) =
1.0

   C(2,3) =
0.0

   C(3,1) =
-2.0

   C(3,2) =
1.0

   C(3,3) =
0.0
```

```
    THE C MATRIX

  0.0         1.00E+00  0.0
  1.00E+00    1.00E+00  0.0
 -2.00E+00    1.00E+00  0.0

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    OBSERVABILITY INDEX =  2

    THE SYSTEM (A,B) IS UNCONTROLLABLE

*******************************************************
   DO YOU WANT TO RUN OBSCON AGAIN?
n

   THIS CONCLUDES THE OBSERVABILITY INDEX/CONTROLLABILITY
   PROGRAM (OBSCON)

   ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
n

   LINCON IS NOW TERMINATED.
```

The complete results presented in Appendix E should be self-explanatory to the user.

C. OPTIMAL CONTROL/KALMAN FILTER PROGRAM (RICATI)

The transient solution, matrix gains, to the Riccati differential equations for the state-regulator controller and the continuous Kalman filter are calculated by the RICATI program.

Given a state-regulator problem with the linear, time-invariant system

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{B}\underline{u}(t) \qquad (4.C\text{-}1)$$

$$\underline{y}(t) = \underline{C}\underline{x}(t) \qquad (4.C\text{-}2)$$

the Riccati equation is defined as

$$\dot{\underline{P}}(t) = -\underline{P}(t)\underline{A}-\underline{A}^T\underline{P}(t)+\underline{P}(t)\underline{B}\underline{R}^{-1}\underline{B}^T\underline{P}(t)-\underline{Q} \qquad (4.C\text{-}3)$$

with $\underline{P}(t_o)$ as the boundary condition. If $\underline{u}(t)$ is not constrained, a gain matrix can be found such that the cost function

$$J=1/2[\underline{x}^T(t_f)\underline{P}(t_f)\underline{x}(t_f)]+1/2\int_{t_o}^{t_f}[\underline{x}^T(t)\underline{Q}\underline{x}(t)+\underline{u}^T(t)\underline{R}\underline{u}(t)]dt \qquad (4.C\text{-}4)$$

is minimized [5]. Such a gain matrix is defined as

$$\underline{G}_c(t) = \underline{R}^{-1}\underline{B}^T\underline{P}(t) \qquad (4.C\text{-}5)$$

and

$$\underline{u}(t) = -\underline{G}_c(t)\underline{x}(t) \tag{4.C-6}$$

The transient solution is solved by the computer. Note that the output from the computer for the gain matrix does not include the negative sign of the feedback loop.

Given a continous Kalman filter problem with the linear, time-invariant system

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{B}\underline{w}(t) \tag{4.C-7}$$

$$\underline{y}(t) = \underline{C}\underline{x}(t) + \underline{v}(t) \tag{4.C-8}$$

where $\underline{w}(t)$, the random process forcing input and $\underline{v}(t)$, the measurement noise, have covariance matrices of $\underline{R}$, the random input covariance matrix, and $\underline{Q}$, the measurement noise covariance matrix, respectfully, the Riccati equation is defined as

$$\dot{\overline{\underline{P}}}(t) = \underline{A}\overline{\underline{P}}(t)+\overline{\underline{P}}(t)\underline{A} + \underline{B}\underline{Q}\underline{B}^T - \overline{\underline{P}}(t)\underline{C}^T\underline{R}^{-1}\underline{C}\overline{\underline{P}}(t) \tag{4.C-9}$$

with

$$\overline{\underline{P}}(t_0) = E\{[\hat{\underline{x}}(t_0) - \overline{\underline{X}}][\hat{\underline{x}}(t_0) - \overline{\underline{X}}]^T\} \tag{4.C-10}$$

as the initial condition boundary. The gain matrix found is defined as

$$\underline{G}_f(t) = \overline{\underline{R}}^{-1}\underline{C}\underline{P}(t) \tag{4.C-11}$$

1. Terminal Session Example

This section contains a session for the specific example presented by Melsa [1]. Commands entered by the user are in lower case. Given the second order linear system

$$\dot{\underline{x}}(t) = \begin{bmatrix} -1.0 & 0.0 \\ 0.5 & 0.0 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \underline{u}(t)$$

$$\underline{y}(t) = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 2.0 \end{bmatrix} \underline{x}(t)$$

determine the optimal transient response control and filter gains for

$$\underline{Q} = \overline{\underline{Q}} = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix}$$

$$\underline{R} = \overline{\underline{R}} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 2.0 \end{bmatrix}$$

For the control option an initial time of 0.0 and a final time of 10.0 is used. For the filter option an initial time of 0.0 and a final time of 5.0 is used. Also, for the filter option, the initial condition matrix is chosen to be

$$\underline{P}(t_o) = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

For both options ten equally spaced values of $\underline{G}$ and $\underline{S}$ on the time interval $t \leq t \leq t$ are used, i.e., NPOINT is set equal to 10.

```
 lincon

 EXECUTION BEGINS...
  LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
    BASIC MATRIX MANIPULATION - <BASMAT>
    RATIONAL TIME RESPONSE - <RTRESP>
    STATE VARIABLE FEEDBACK - <STVAR>
    CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
    LUENBERGER OBSERVER - <LUEN>
    OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
    DISCRETE TIME KALMAN FILTER - <KALMAN>
    OPTIMAL CONTROL - <OPTCON>
    PARTIAL FRACTION EXPANSION - <PRFEXP>
    ROOTS OF A POLYNOMIAL - <ROOTS>
 TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
 SYMBOLS < >.
 ricati

    RICATI DETERMINES THE TRANSIENT SOLUTION FOR THE RICCATI
    EQUATION. FIRST ENTER THE PROBLEM IDENTIFICATION
    (*NOT TO EXCEED 20 CHARACTERS*)
 thesis example

    NOW, ENTER THE ORDER OF THE SYSTEM (UP TO 8).
 2

    ENTER THE ELEMENTS OF THE PLANT MATRIX--A.

    A(1,1) =
 -1.0

    A(1,2) =
 0.0

    A(2,1) =
 0.0

    A(2,2) =
 -2.0

     THE A  MATRIX (PLANT MATRIX)
 -1.00E+00  0.0
  0.0       -2.00E+00


    DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
 n

    ENTER THE NUMBER OF CONTROL INPUTS.
 2
```

```
   ENTER THE ELEMENTS OF THE DISTRIBUTION MATRIX--B.
   B(1,1) =
1.0
   B(1,2) =
0.0
   B(2,1) =
0.0
   B(2,2) =
1.0
    THE B  MATRIX (DISTRIBUTION MATRIX)

 1.00E+00  0.0
 0.0       1.00E+00
   DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
   ENTER THE NUMBER OF OBSERVABLE OUTPUTS.
2
   ENTER THE ELEMENTS OF THE MEASUREMENT MATRIX--C.
   C(1,1) =
1.0
   C(1,2) =
0.0
   C(2,1) =
0.0
   C(2,2) =
2.0
    THE C  MATRIX (MEASUREMENT MATRIX)

1.00E+00  0.0
0.0       2.00E+00
   DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
   YOU HAVE TWO OPTIONS AVAILABLE:
      (1)  THE CONTROL OPTION FOR SOLVING STATE-REGULATOR
           PROBLEMS, OR
      (2)  THE FILTER OPTION FOR SOLVING A CONTINUOUS KALMAN
           FILTER PROBLEM.
   FOR THE CONTROL OPTION, ENTER THE LETTER C.
   FOR THE FILTER OPTION, ENTER THE LETTER F.
c
   ENTER THE ELEMENTS OF THE CONTROL WEIGHTING MATRIX--R.
   R(1,1) =
1.0
   R(1,2) =
0.0
   R(2,1) =
0.0
   R(2,2) =
2.0
```

```
     THE R MATRIX (CONTROL WEIGHTING MATRIX)

  1.00E+00   0.0
  0.0        2.00E+00

    DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    ENTER THE ELEMENTS OF THE STATE WEIGHTING MATRIX--Q.

    Q(1,1) =
1.0
    Q(1,2) =
1.0
    Q(2,1) =
1.0
    Q(2,2) =
1.0
     THE Q MATRIX (STATE WEIGHTING MATRIX)

  1.00E+00   1.00E+00
  1.00E+00   1.00E+00

    DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    ENTER THE INITIAL TIME FOR THE TRANSIENT RESPONSE.
0.0

    ENTER THE FINAL TIME FOR THE TRANSIENT RESPONSE.
10.0

    ENTER THE NUMBER OF POINTS OF THE TRANSIENT RESPONSE
    TO BE PRINTED.    (<100)
10

    **********************
    *** CONTROL OPTION ***
    **********************
    ENTER THE ELEMENTS OF THE TERMINAL BOUNDARY VALUE
    MATRIX--P.

    P(1,1) =
0.0
    P(1,2) =
0.0
    P(2,1) =
0.0
    P(2,2) =
0.0
     THE P MATRIX (TERMINAL BOUNDARY VALUE MATRIX)

  0.0        0.0
  0.0        0.0

    DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
```

```
     TRANSIENT SOLUTION

    TIME =   1.000E+01
     GAINS
 0.0         0.0
 0.0         0.0

    TIME =   9.000E+00
     GAINS
 3.75E-01    2.81E-01
 1.40E-01    1.11E-01

    TIME =   8.000E+00
     GAINS
 3.98E-01    2.85E-01
 1.42E-01    1.12E-01

    TIME =   7.000E+00
     GAINS
 4.00E-01    2.85E-01
 1.42E-01    1.12E-01

    TIME =   6.000E+00
     GAINS
 4.00E-01    2.85E-01
 1.42E-01    1.12E-01

    TIME =   5.000E+00
     GAINS
 4.00E-01    2.85E-01
 1.42E-01    1.12E-01

    TIME =   4.000E+00
     GAINS
 4.00E-01    2.85E-01
 1.42E-01    1.12E-01

    TIME =   3.000E+00
     GAINS
 4.00E-01    2.85E-01
 1.42E-01    1.12E-01

    TIME =   2.000E+00
     GAINS
 4.00E-01    2.85E-01
 1.42E-01    1.12E-01

    TIME =   1.000E+00
     GAINS
 4.00E-01    2.85E-01
 1.42E-01    1.12E-01

    TIME =   2.861E-06
     GAINS
 4.00E-01    2.85E-01
 1.42E-01    1.12E-01

************************************************************
```

```
   DO YOU WANT THE FILTER OPTION?
Y
   WILL THERE BE CHANGES TO THE A, B, OR C MATRICES?
n
   WILL THERE BE CHANGES TO THE R OR Q MATRICES?
n
   ENTER THE INITIAL TIME FOR THE TRANSIENT RESPONSE.
0.0
   ENTER THE FINAL TIME FOR THE TRANSIENT RESPONSE.
5.0
   ENTER THE NUMBER OF POINTS OF THE TRANSIENT RESPONSE TO
   BE PRINTED. (<100)
10
   **********************
   *** FILTER OPTION ***
   **********************
   ENTER THE ELEMENTS OF THE INITIAL BOUNDARY VALUE
   MATRIX--P.
   P(1,1)=
0.0
   P(1,2)=
0.0
   P(2,1)=
0.0
   P(2,2)=
0.0
    THE P MATRIX (INITIAL BOUNDARY VALUE MATRIX)
 0.0        0.0
 0.0        0.0
   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
    TRANSIENT SOLUTION
   TIME =   0.0
    GAINS
 0.0        0.0
 0.0        0.0

   TIME =   5.000E-01
    GAINS
 2.39E-01   2.00E-01
 8.02E-01   6.85E-01

   TIME =   1.000E+00
    GAINS
 2.71E-01   2.13E-01
 8.51E-01   7.05E-01

   TIME =   1.500E+00
    GAINS
 2.77E-01   2.13E-01
 8.53E-01   7.06E-01
```

```
    TIME =   2.000E+00
     GAINS
  2.78E-01   2.13E-01
  8.53E-01   7.06E-01

    TIME =   2.500E+00
     GAINS
  2.79E-01   2.13E-01
  8.53E-01   7.06E-01

    TIME =   3.000E+00
     GAINS
  2.79E-01   2.13E-01
  8.53E-01   7.06E-01

    TIME =   3.500E+00
     GAINS
  2.79E-01   2.13E-01
  8.53E-01   7.06E-01

    TIME =   4.000E+00
     GAINS
  2.79E-01   2.13E-01
  8.53E-01   7.06E-01

    TIME =   4.500E+00
     GAINS
  2.79E-01   2.13E-01
  8.53E-01   7.06E-01

    TIME =   5.000E+00
     GAINS
  2.79E-01   2.13E-01
  8.53E-01   7.06E-01
*****************************************************************
    DO YOU WANT THE CONTROL OPTION?
 n
    THIS CONCLUDES THE RICCATI EQUATION PROGRAM (RICATI).
    ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
 n
    LINCON IS NOW TERMINATED.
```

The computer results for both options presented in Appendix F indicates that a steady state gain matrix has been achieved.

D. DISCRETE TIME KALMAN FILTER.(KALMAN)

The discrete Kalman filter gain matrix, $\underline{G}(k)$, is calculated by this program. The theoretical concepts involved in the development of the computer codes are described by Sage [6]. A brief development of the discrete Kalman filter is included here as an aid in the use of KALMAN. Additional reference materials can be found in the bibliography.

Kalman filtering is a method of obtaining minimum-variance estimates of signals from noisy measurements. The discrete Kalman filter provides state estimates for the following system

$$\underline{x}(k) = \underline{\Phi}\,\underline{x}(k-1)+\underline{\Delta}\,\underline{u}(k-1)+\underline{\Gamma}\,\underline{w}(k-1) \qquad (4.D-1)$$

with the discrete linear observations

$$\underline{z}(k) = \underline{H}\,\underline{x}(k)+\underline{v}(k) \qquad (4.D-2)$$

where

$\underline{x}$ = the nx1 state vector at the time t(k)

$\underline{\Phi}$ = the nxn nonsingular state transition matrix

$\underline{\Gamma}$ = the nxr disturbance transition or distribution matrix

$\underline{\Delta}$ = the nxp control distribution matrix

$\underline{w}$ = the rx1 disturbance of system random input vector

$\underline{z}$ = the mx1 measurement vector

$\underline{H}$ = the mx1 measurement or observation vector

$\underline{v}$ = the mx1 measurement noise vector

$\underline{u}$ = the px1 control or test signal vector

k = the discrete-time index (k = 0,1,...)

The optimal filtered estimate of $\underline{x}(k)$, denoted $\hat{\underline{x}}(k|k)$, is given by the recursive relations

$$\hat{\underline{x}}(k|k-1) = \underline{\Phi}\underline{x}(k-1|k-1) + \underline{\psi}\underline{u}(k-1) \tag{4.D-3}$$

and

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + \underline{G}(k)[\underline{z}(k)-\underline{H}\hat{\underline{x}}(k|k-1)] \tag{4.D-4}$$

for k = 0,1,..., where $\underline{x}(0,0) = \underline{0}$. The Kalman gain matrix, $\underline{G}(k)$, is an nxm matrix which is specified as

$$\underline{G}(k)=\underline{P}(k|k-1)\underline{H}^T[\underline{H}\underline{P}(k|k-1)\underline{H}^T+\underline{R}(k)]^{-1} \tag{4.D-5}$$

$$\underline{P}(k|k) = [\underline{I}-\underline{G}(k)\underline{H}]\underline{P}(k|k-1) \tag{4.D-6}$$

$$\underline{P}(k|k-1) = \underline{\Phi}\underline{P}(k-1|k-1)\underline{\Phi}^T+\underline{Q}(k,k-1) \tag{4.D-7}$$

where

$\underline{I}$ = the nxn identity matrix

$\underline{P}(0|0) = \underline{P}(0)$, the initial condition matrix

$\hat{\underline{x}}(k|k-1)$ = the single-stage prediction of $\underline{x}(k)$

$\hat{\underline{x}}(k|k)$ = the filtered estimate of $\underline{x}(k)$

$\underline{Q} = \underline{\Gamma} E\{\underline{w}(k)\underline{w}(k)^T\}\underline{\Gamma}^T$, the nxn covariance matrix of the random input

$\underline{R}$ = the mxn covariance matrix of the measurement noise

$E\{\underline{w}(k)\underline{w}(k)^T\}$, the mean-square magnitude of the perturbation acceleration matrix

1. Input Requirements

This program computes the recurrence Eqs. (4.D-5),(4.D-6) and (4.D-7) for a specified number of iterations NP and prints $\underline{G}(k)$ as a function of k.

The required inputs are

1) the transition matrix ($\underline{\Phi}$)

2) the distribution matrix ($\underline{\Gamma}$)

3) $E\{\underline{w}(k)\underline{w}(k)^T\}$

4) the observation matrix ($\underline{H}$)

5) the number of points to be performed (NP)

and

6) the initial condition matrix $[\underline{P}(0|0)]$.

2. Terminal Session Example

This section contains a session for a specific problem. Commands entered by the user are in lower case. The following system is to be tested:

$$\underline{x}(k) = \begin{bmatrix} 1.0 & 0.5 \\ 0.0 & 1.0 \end{bmatrix} \underline{x}(k-1) + \begin{bmatrix} 0.125 \\ 0.5 \end{bmatrix} \underline{w}(k-1)$$

$$\underline{z}(k) = [\,1.0 \quad 0.0\,]\ \underline{x}(k) + \underline{v}(k)$$

$$\underline{R} = [5.0]$$

$$E\{\underline{w}(k)\underline{w}(k)^T\} = 4.0$$

$$\underline{P}(0) = \begin{bmatrix} 1000.0 & 0.0 \\ 0.0 & 1000.0 \end{bmatrix}$$

The number of time points to be computed is chosen to be 20.

```
lincon

EXECUTION BEGINS...
 LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
   BASIC MATRIX MANIPULATION - <BASMAT>
   RATIONAL TIME RESPONSE - <RTRESP>
   STATE VARIABLE FEEDBACK - <STVAR>
   CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
   LUENBERGER OBSERVER - <LUEN>
   OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
   DISCRETE TIME KALMAN FILTER - <KALMAN>
   OPTIMAL CONTROL - <OPTCON>
   PARTIAL FRACTION EXPANSION - <PRFEXP>
   ROOTS OF A POLYNOMIAL - <ROOTS>
 TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
 SYMBOLS < >.
kalman

   KALMAN DETERMINES THE DISCRETE KALMAN FILTER GAIN MATRIX
   --G(K). FIRST ENTER THE PROBLEM IDENTIFICATION
   (*NOT TO EXCEED 20 CHARACTERS*).
thesis example

   NOW, ENTER THE ORDER OF THE SYSTEM (UP TO 8).
2
```

```
   ENTER THE ELEMENTS OF THE TRANSITON MATRIX--PHI
   PHI(1,1)=
1.

   PHI(1,2)=
.5

   PHI(2,1)=
0.0

   PHI(2,2)=
1.

    THE PHI MATRIX (TRANSITION MATRIX)

   1.00E+00      5.00E-01
   0.0           1.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE DIMENSION OF THE RANDOM INPUT VECTOR.
1

   ENTER THE ELEMENTS OF THE DISTRIBUTION MATRIX--GAMMA.
   GAMMA(1,1) =
.125

   GAMMA(2,1) =
.5

    THE GAMMA MATRIX (DISTRIBUTION MATRIX)

   1.25E-01
   5.00E-01

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE ELEMENTS OF THE MEAN-SQUARE MAGNITUDE OF THE
   PERTURBATION ACCELERATION MATRIX--W.

   W(1,1) =
4.

    THE W MATRIX (MEAN-SQUARE MAGNITUDE OF THE
    PERTURBATION ACCELERATION MATRIX)

   4.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE NUMBER OF OUTPUTS.
1

   ENTER THE ELEMENTS OF THE OBSERVATION MATRIX--H.

   H(1,1) =
1.

   H(1,2) =
0.

    THE H MATRIX (OBSERVATION MATRIX)

   1.00E+00      0.0
```

```
   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE ELEMENTS OF THE MEASUREMENT NOISE COVARIANCE
   MATRIX--R.

   R(1,1) =
5.

    THE R MATRIX (MEASUREMENT NOISE COVARIANCE MATRIX)

   5.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE NUMBER OF THE POINTS TO BE PERFORMED.
       (<100)
20

   ENTER THE ELEMENTS OF THE INITIAL CONDITION MATRIX--P.

   P(1,1) =
1000.

   P(1,2) =
0.

   P(2,1) =
0.

   P(2,2) =
1000.

    THE P MATRIX (INITIAL CONDITION MATRIX)

   1.00E+03     0.0
   0.0          1.00E+03
   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

     K =    0
    GAINS
   9.95E-01     0.0


     K =    1
    GAINS
   9.81E-01     1.92E+00


     K =    2
    GAINS
   8.29E-01     9.98E-01


     K =    3
    GAINS
   7.03E-01     6.24E-01


     K =    4
    GAINS
   6.13E-01     4.52E-01
```

```
  K =   5
 GAINS
5.54E-01      3.72E-01

  K =   6
 GAINS
5.18E-01      3.35E-01

  K =   7
 GAINS
4.99E-01      3.23E-01

  K =   8
 GAINS
4.90E-01      3.20E-01

  K =   9
 GAINS
4.87E-01      3.20E-01

  K =  10
 GAINS
4.86E-01      3.21E-01

  K =  11
 GAINS
4.86E-01      3.21E-01

  K =  12
 GAINS
4.86E-01      3.21E-01

  K =  13
 GAINS
4.86E-01      3.21E-01

  K =  14
 GAINS
4.86E-01      3.21E-01

  K =  15
 GAINS
4.86E-01      3.21E-01

  K =  16
 GAINS
4.86E-01      3.21E-01

  K =  17
 GAINS
4.86E-01      3.21E-01

  K =  18
 GAINS
4.86E-01      3.21E-01
```

```
   K =   19
  GAINS
4.86E-01     3.21E-01

   K =   20
  GAINS
4.86E-01     3.21E-01

THIS CONCLUDES THE DISCRETE KALMAN FILTER PROGRAM
(KALMAN).

ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
n
LINCON IS NOW TERMINATED.
```

The computer results are presented in Appendix G.

E. STATE VARIABLE FEEDBACK PROGRAM (STVAR)

Given the linear time-invariant system

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{b}u(t) \qquad (4.E-1)$$

$$u(t) = K[r(t) - \underline{k}^T\underline{x}(t)] \qquad (4.E-2)$$

$$y(t) = \underline{c}^T\underline{x}(t) \qquad (4.E.3)$$

the following can be performed by STVAR

(1) calculation of the plant tranfer function, $Y(s)/U(s)$

(2) by defining a fictitious $\underline{c}$ vector the internal transfer function can be calculated, $X_i(s)/U(s)$; for example, if $X_3(s)/U(s)$ is desired the $\underline{c}$ matrix is selected with $c_3 = 1$ and all other $\underline{c}$ elements equal to zero; or if $X_1(s)/X_3(s)$ is desired, calculate $X_1(s)/U(s)$ and $X_3(s)/U(s)$ and divide the two

(3) calculation of the closed-loop transfer function, $Y(s)/R(s)$

(4) calculation of the feedback transfer function, $H_{eq}(s)$

(5) for a desired closed-loop transfer function, the controller gain and feedback coefficients in addition to $H_{eq}(s)$ can be calculated; the

feedforward gain is selected so that a zero steady state error results from a step input; the designer who wishes other conditions must rescale the gain and feedback coefficients appropriately; for example, if it is desired to have $K = 1.0$ but it is calculated as $K = 2.0$ with feedback coefficients of $k = 0.5$, $k = 0.0$ and $k = 1.5$, the procedure to modify the results would be

$$\underline{k} = \frac{2.0}{1.0} [0.5 \quad 0.0 \quad 1.5]$$

$$= [1.0 \quad 0.0 \quad 3.0]$$

All of the information necessary for the user to solve state variable feedback problems is presented in this section. However, the theoretical concepts involved in the development of the computer codes are fully described by Melsa [7].

The basic input contains the problem identification, matrices $\underline{A}$ and $\underline{b}$ and the order of the plant, n. These four inputs are required regardless of what open- or closed-loop calculations are to be made.

At this point STVAR verifies the controllability of the system. Three controllability conditions are possible

(1) complete controllability

(2) numerically uncontrollable

and

(3) uncontrollability

Contollability arises when the controllability matrix

$$\underline{E} = [\ \underline{b}\ \underline{A}\underline{b}\ \underline{A}^2\underline{b}\ \ldots\ \underline{A}^{n-1}\underline{b}] \qquad (4.E-4)$$

is nonsingular, i.e., $\det\underline{E} \neq 0$. Even if the matrix is nonsingular problems may arise if it is difficult to invert. To check this STVAR multiplies the matrix by its calculated inverse. The result should be the identity matrix. The actual matrix product is compared with the identity matrix to provide a measure of uncontrollability. If the maximum value of deviation is not negligible, the plant is identified as numerically uncontrollable. A deviation larger than 10^{-3} to 10^{-5} has been found to indicate difficulty by Melsa [1].

User beware: If the system is identified as being uncontrollable, all open- and closed-loop calculations are still performed!

There are three possible closed-loop options available, one for analysis purposes only and the other two for design purposes. After choosing the analysis option, and supplying STVAR with the feedforward gain K and feedback coefficient

matrix $\underline{k}$, the program calculates the closed-loop characteristic polynomial and the numerator of the equivalent feedback transfer function.

The two design options are used to calculate the controller gain and the feedback coefficients necessary for a desired closed-loop characteristic polynomial. The polynomial may be entered in either polynomial form or factored form.

1. Terminal Session Example

This section contains a third order system for analysis by STVAR as presented by Melsa [1]. The state variable representation of the plant is given by

$$\dot{\underline{x}}(t) = \begin{bmatrix} -1.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & -3.0 & 0.0 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} \underline{u}(t)$$

$$\underline{y}(t) = [\,1.0 \quad 1.0 \quad 0.0\,]\, \underline{x}(t)$$

For the open-loop case, it is desired to find the internal transfer function X (s)/U (s) and the plant transfer function Y (s)/U (s). A ficticious matrix, necessary to find the internal transfer function, is then

$$\underline{c} = [0.0 \quad 0.0 \quad 1.0]$$

In addition, find the values of the feedforward gain and the feedback coefficients required to give a closed-loop transfer function of

$$\frac{Y(s)}{R(s)} = \frac{2(s+2)}{s^3+4s^2+6s+4}$$

```
lincon

EXECUTION BEGINS...
LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
    BASIC MATRIX MANIPULATION - <BASMAT>
    RATIONAL TIME RESPONSE - <RTRESP>
    STATE VARIABLE FEEDBACK - <STVAR>
    CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
    LUENBERGER OBSERVER - <LUEN>
    OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
    DISCRETE TIME KALMAN FILTER - <KALMAN>
    OPTIMAL CONTROL - <OPTCON>
    PARTIAL FRACTION EXPANSION - <PRFEXP>
    ROOTS OF A POLYNOMIAL - <ROOTS>
TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
SYMBOLS < >.
stvar

    STVAR DETERMINES INTERNAL TRANSFER FUNCTIONS,
    THE PLANT TRANSFER FUNCTION, THE CLOSED-LOOP TRANSFER
    FUNCTION, AND THE EQUIVALENT FEEDBACK TRANSFER FUNCTION.
    IN ADDITION, THE CONTROLLER GAIN AND THE FEEDBACK
    COEFFICIENTS NECESSARY TO ACHIEVE A SPECIFIED CLOSED-
    LOOP TRANSFER FUNCTION ARE CALCULATED.

    FIRST ENTER THE PROBLEM IDENTIFICATION
     (*NOT TO EXCEED 20 CHARACTERS*).
thesis example

    NOW, ENTER THE ORDER OF THE SYSTEM (UP TO 8).
3

    ENTER THE ELEMENTS OF THE PLANT MATRIX--A
    A(1,1)=
-1.

    A(1,2)=
1.

    A(1,3)=
0

    A(2,1)=
0

    A(2,2)=
0

    A(2,3)=
1.

    A(3,1)=
0
```

```
   A(3,2)=
-3.
   A(3,3)=
0
    THE A  MATRIX (PLANT MATRIX)
       -1.00E+00   1.00E+00   0.0
        0.0        0.0        1.00E+00
        0.0       -3.00E+00   0.0

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
   ENTER THE ELEMENTS OF THE CONTROL VECTOR--B.
   B(1)=
0
   B(2)=
0
   B(3)=
1.
    THE B  MATRIX (CONTROL MATRIX)
        0.0
        0.0
        1.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
*******************************************************************
    OPEN-LOOP CALCULATIONS

    DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
        3.00E+00   3.00E+00   1.00E+00   1.00E+00

    THE ROOTS ARE          REAL PART  IMAG. PART
                            0.0       -1.73E+00
                            0.0        1.73E+00
                           -1.00E+00   0.0

   DO YOU HAVE A FICTICIOUS OUTPUT VECTOR TO ENTER?
y
   ENTER THE ELEMENTS OF THE FICTICIOUS OUTPUT VECTOR--C.
   C(1)=
0
   C(2)=
0
   C(3)=
1.
    THE C  MATRIX (FICTICIOUS OUTPUT VECTOR)
        0.0
        0.0
        1.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
    NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S

        0.0        1.00E+00   1.00E+00
```

```
    THE ROOTS ARE       REAL PART IMAG. PART
                        -1.00E+00  0.0
                         0.0       0.0
   DO YOU HAVE ANOTHER FICTICIOUS MATRIX TO ENTER?
n

   ENTER THE ELEMENTS OF THE TRUE OUTPUT VECTOR--C.
   C(1)=
1.

   C(2)=
1.

   C(3)=
0

    THE C MATRIX (OUTPUT VECTOR)
        1.00E+00
        1.00E+00
        0.0

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
        2.00E+00  1.00E+00

    THE ROOTS ARE       REAL PART IMAG. PART
                        -2.00E+00  0.0
   THIS PROGRAM IS CAPABLE OF PERFORMING THREE TYPES
   OF CLOSED-LOOP CALCULATIONS. ONE TYPE FOR THE ANALYSIS
   MODE. THE OTHER TWO FOR DESIGN.

   ENTER ONE OF THE FOLLOWING MODES:
      1) A -- FOR THE ANALYSIS MODE
      2) P -- FOR THE DESIGN MODE WITH THE UNFACTORED
              CLOSED-LOOP CHARACTERISTIC POLYNOMIAL
      3) F -- FOR THE DESIGN MODE WITH THE FACTORED
              CLOSED-LOOP CHARACTERISTIC POLYNOMIAL
p

    CLOSED-LOOP CALCULATIONS

    KEY = P   *****
   ENTER THE DESIRED CLOSED-LOOP CHARACTERISTIC POLYNOMIAL
   COEFFICIENTS IN ASCENDING POWERS OF S.
   YOUR HIGHEST ORDER COEFFICIENT MUST BE A VALUE OF ONE.
   POLY(1)=
4.

   POLY(2)=
6.

   POLY(3)=
4.

   POLY(4)=
1.

    THE NUMERATOR OF H-EQUIVALENT -
    IN ASCENDING POWERS OF S
        5.00E-01  1.50E+00  1.50E+00

    THE ROOTS ARE       REAL PART IMAG. PART
                        -5.00E-01 -2.89E-01
                        -5.00E-01  2.89E-01
```

```
    THE FEEDBACK COEFFICIENTS
        5.00E-01  0.0        1.50E+00

    THE GAIN =    2.000000E+00

    THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL -
    IN ASCENDING POWERS OF S
        4.00E+00  6.00E+00  4.00E+00  1.00E+00

    THE ROOTS ARE         REAL PART IMAG. PART
                          -2.00E+00  0.0
                          -1.00E+00 -1.00E+00
                          -1.00E+00  1.00E+00

    MAXIMUM NORMALIZED ERROR =  0.0

   DO YOU WANT TO RUN ANOTHER MODE IN STVAR?
n

   THIS CONCLUDES THE STATE VARIABLE FEEDBACK PROGRAM
   (STVAR).

   ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
n

   LINCON IS NOW TERMINATED.
```

The computer results for this problem is presented in Appendix H. The open-loop portion indicates that the system is controllable, since there is no indication of uncontrollability, with the desired internal transfer function of

$$\frac{X_3(s)}{U(s)} = \frac{s^2+s}{s^3+s^2+3s+3}$$

and a plant transfer function of

$$\frac{Y(s)}{U(s)} = \frac{s+2}{s^3+s^2+3s+3}$$

The closed-loop portion, using the polynomial design mode, shows that the feedforward gain and feedback

coefficient matrix required to yield the desired closed-loop transfer function, Y(s)/R(s), are

$$K = 2.0$$

and

$$\underline{k} = \begin{bmatrix} 0.5 \\ 0.0 \\ 1.5 \end{bmatrix}$$

F. LUENBERGER OBSERVER PROGRAM (LUEN)

When a particular closed-loop transfer function is desired and some of the states are inaccessible, LUEN can be used to design a Luenberger Observer. For example, if q measurements are state variables, an observer of reduced dimensions can be designed to estimate only those states which are not measured. The state estimates generated by an observer can be used as input information to a controller [8]. The block diagram presented in Fig. 4-1 represents the general form of the system when a compensator is placed in the feedback path.

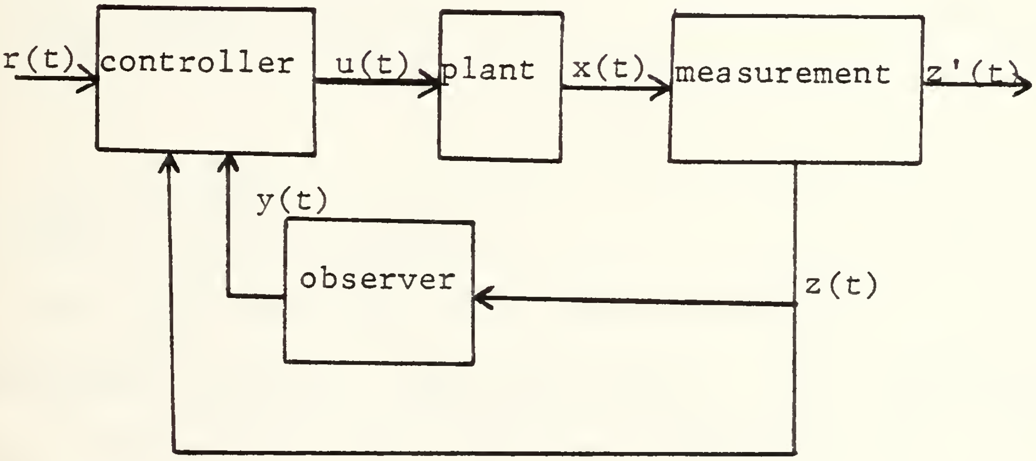

Fig. 4.1. Luenberger Observer Block Diagram

The plant is characterized by the state and measurement equations

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{B}u(t) \tag{4.F-1}$$

$$\underline{z}(t) = \underline{C}\underline{x}(t) \tag{4.F-2}$$

$$z'(t) = \underline{c}\ \underline{x}(t) \tag{4.F-3}$$

An observer can be designed that generates an estimate $\underline{y}$ which converges to the state $\underline{x}$ as time becomes large [9]. A linear controller is designed as

$$\underline{u}(t) = K[\underline{r}(t) - \underline{k}^T\underline{x}(t)] \tag{4.F-4}$$

in which all states can be measured. Replacing the true state with the estimate yields

$$\underline{u}(t) = K[\underline{r}(t) - \underline{k}^T\underline{y}(t)] \tag{4.F-5}$$

where

$$\underline{k}^T\underline{y}(t) = \underline{h}^T\underline{y}(t) + \underline{g}^T\underline{z}(t) \tag{4.F-6}$$

As time increases $\underline{k}^T\underline{y}(t)$ will approach $\underline{k}^T\underline{x}(t)$. Since the reduced-order observer estimates only the inaccesible states, the control law contains measured states where available and observer estimates for the other states.

For ease in using LUEN, the following are defined:

$\underline{x}(t)$ = n-element column state vector

$\underline{u}(t)$ = plant input

$\underline{z}(t)$ = q-vector of system measurements

$z'(t)$ = output variable to be controlled

$\underline{A}$ = plant matrix (n x n)

$\underline{b}$ = distribution matrix (q x m)

[assumptions: (1) $q \leq n$ and (2) $\underline{C}$ has rank q]

$\underline{C}$ = output matrix

$r(t)$ = forcing function

K = feedforward or controller gain

$\underline{k}$ = feedback coefficient matrix(n x m)

$\underline{h}$ = observer feedback coefficient matrix

$\underline{g}$ = output feedback coefficient matrix

and

$\underline{y}(t)$ = estimated state vector of $\underline{x}(t)$.

From theory, the observer is described by

$$\underline{y}(t) = \underline{F}\underline{y}(t) + \underline{G}_1\underline{z}(t) + \underline{G}_2 u(t) \qquad (4.F-7)$$

where

$\underline{F}$ = observer eigenvalue matrix

and

$\underline{G}_1$ and $\underline{G}_2$ = observer gain matrices.

Z'(s)/R(s) can be solved by the following procedure:

(1) select Z'(s)/R(s) and solve for K and $\underline{k}$ by using the STVAR program

(2) use the OBSCON program to calculate the observability index; the observer is designed to have a dimension greater than or equal to the observability index minus one.

(3) select the $\underline{F}$ eigenvalues; these should not equal those of $\underline{A}$, previously calculated by STVAR.

(4) use LUEN to calculate $\underline{G}_1$, $\underline{G}_2$, $\underline{h}$ and $\underline{g}$.

1. Terminal Session Example

The example presented here is taken from Desjardins, pp.147 [2]. Commands entered by the user are in lower

case. However, due to the special nature of the example, comments have been added to the terminal session for clarity. Given the fourth order plant

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & -15. & -23. & -9.0 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} u(t)$$

with

$$z'(t) = [10. \quad 20. \quad 0.0 \quad 0.0]\, \underline{x}(t)$$

Because x_1 and x_2 are the only measureable states

$$\underline{y}(t) = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \end{bmatrix} \underline{x}(t)$$

The closed-loop transfer function to be achieved is chosen to be

$$\frac{Z'(s)}{R(s)} = \frac{1}{s^4+5s^3+17s^2+28s+20}$$

```
lincon

EXECUTION BEGINS...
LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
   BASIC MATRIX MANIPULATION - <BASMAT>
   RATIONAL TIME RESPONSE - <RTRESP>
   STATE VARIABLE FEEDBACK - <STVAR>
   CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
   LUENBERGER OBSERVER - <LUEN>
   OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
   DISCRETE TIME KALMAN FILTER - <KALMAN>
   OPTIMAL CONTROL - <OPTCON>
   PARTIAL FRACTION EXPANSION - <PRFEXP>
```

```
   ROOTS OF A POLYNOMIAL - <ROOTS>
TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
SYMBOLS < >.
stvar

   STVAR DETERMINES INTERNAL TRANSFER FUNCTIONS,
   THE PLANT TRANSFER FUNCTION, THE CLOSED-LOOP TRANSFER
   FUNCTION, AND THE EQUIVALENT FEEDBACK TRANSFER FUNCTION.
   IN ADDITION, THE CONTROLLER GAIN AND THE FEEDBACK
   COEFFICIENTS NECESSARY TO ACHIEVE A SPECIFIED CLOSED-
   LOOP TRANSFER FUNCTION ARE CALCULATED.

   FIRST ENTER THE PROBLEM IDENTIFICATION
   (*NOT TO EXCEED 20 CHARACTERS*).
THESIS EXAMPLE

   NOW, ENTER THE ORDER OF THE SYSTEM (UP TO 8).
4

   ENTER THE ELEMENTS OF THE PLANT MATRIX--A

   A(1,1) =
0

   A(1,2) =
1.

   A(1,3) =
0

   A(1,4) =
0

   A(2,1) =
0

   A(2,2) =
0

   A(2,3) =
1.

   A(2,4) =
0

   A(3,1) =
0

   A(3,2) =
0

   A(3,3) =
0

   A(3,4) =
1.

   A(4,1) =
0

   A(4,2) =
-15.

   A(4,3) =
-23.

   A(4,4) =
-9.
```

```
     THE A  MATRIX (PLANT MATRIX)

         0.0         1.00E+00  0.0         0.0
         0.0         0.0         1.00E+00  0.0
         0.0         0.0         0.0         1.00E+00
         0.0        -1.50E+01 -2.30E+01 -9.00E+00

    DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    ENTER THE ELEMENTS OF THE CONTROL VECTOR--B.

    B(1) =
0

    B(2) =
0

    B(3) =
0

    B(4) =
1.

     THE B  MATRIX (CONTROL MATRIX)

         0.0
         0.0
         0.0
         1.00E+00

    DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
*****************************************************************

     OPEN-LOOP CALCULATIONS

     DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S

         0.0         1.50E+01  2.30E+01  9.00E+00  1.00E+00

     THE ROOTS ARE          REAL PART IMAG. PART
                            -3.00E+00  0.0
                            -5.00E+00  0.0
                            -1.00E+00  0.0
                             0.0       0.0

    DO YOU HAVE A FICTICIOUS OUTPUT VECTOR TO ENTER?
n

    ENTER THE ELEMENTS OF THE TRUE OUTPUT VECTOR--C.

    C(1) =
20.

    C(2) =
10.

    C(3) =
0

    C(4) =

     THE C  MATRIX (OUTPUT VECTOR)
         2.00E+01
         1.00E+01
         0.0
         0.0
```

```
   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S

        2.00E+01  1.00E+01

    THE ROOTS ARE        REAL PART IMAG. PART
                         -2.00E+00  0.0

   THIS PROGRAM IS CAPABLE OF PERFORMING THREE TYPES
   OF CLOSED-LOOP CALCULATIONS. ONE TYPE FOR THE ANALYSIS
   MODE. THE OTHER TWO FOR DESIGN.

   ENTER ONE OF THE FOLLOWING MODES:
      1) A -- FOR THE ANALYSIS MODE
      2) P -- FOR THE DESIGN MODE WITH THE UNFACTORED
              CLOSED-LOOP CHARACTERISTIC POLYNOMIAL
      3) F -- FOR THE DESIGN MODE WITH THE FACTORED
              CLOSED-LOOP CHARACTERISTIC POLYNOMIAL
p

    CLOSED-LOOP CALCULATIONS

    KEY = P    *****
   ENTER THE DESIRED CLOSED-LOOP CHARACTERISTIC POLYNOMIAL
   COEFFICIENTS IN ASCENDING POWERS OF S.

   YOUR HIGHEST ORDER COEFFICIENT MUST BE A VALUE OF ONE.
   POLY(1) =
20.

   POLY(2) =
28.

   POLY(3) =
17.

   POLY(4) =
6.

   POLY(5) =
1.

    THE NUMERATOR OF H-EQUIVALENT -
    IN ASCENDING POWERS OF S

        2.00E+01  1.30E+01 -6.00E+00 -3.00E+00

    THE ROOTS ARE        REAL PART IMAG. PART
                         -1.04E+00  0.0
                          3.21E+00  0.0

    THE FEEDBACK COEFFICIENTS

        2.00E+01  1.30E+01 -6.00E+00 -3.00E+00

    THE GAIN =     1.000000E+00

    THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL -
    IN ASCENDING POWERS OF S

        2.00E+01  2.30E+01  1.70E+01  6.00E+00  1.00E+00
```

```
    THE ROOTS ARE          REAL PART  IMAG. PART
                           -1.00E+00  -2.00E+00
                           -1.00E+00   2.00E+00
                           -2.00E+00   0.0
                           -2.00E+00   0.0

    MAXIMUM NORMALIZED ERROR =  0.0

   DO YOU WANT TO RUN ANOTHER MODE IN STVAR?
n

   THIS CONCLUDES THE STATE VARIABLE FEEDBACK PROGRAM
   (STVAR).

      COMMENT:  results  indicate  that  the  system  is
      controllable,  that  the eigenvalues of A  are -3.0,
      -5.0,  -1.0 and 0.0,  that the values of k are 20.0,
      13.0,  -6.0 and  -3.0 and that K is  equal to unity;
      the observability index is next needed to design the
      observer


   ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
y

LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
   BASIC MATRIX MANIPULATION - <BASMAT>
   RATIONAL TIME RESPONSE - <RTRESP>
   STATE VARIABLE FEEDBACK - <STVAR>
   CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
   LUENBERGER OBSERVER - <LUEN>
   OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
   DISCRETE TIME KALMAN FILTER - <KALMAN>
   OPTIMAL CONTROL - <OPTCON>
   PARTIAL FRACTION EXPANSION - <PRFEXP>
   ROOTS OF A POLYNOMIAL - <ROOTS>
TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
SYMBOLS < >.
obscon

   OBSCON DETERMINES THE OBSERVABILITY INDEX AND
   CONTROLLABILITY OF A SYSTEM.

   FIRST ENTER THE PROBLEM IDENTIFICATION
   (*NOT TO EXCEED 20 CHARACTERS*).
thesis example

   NOW, ENTER THE ORDER OF THE SYSTEM (UP TO 8).
4

   ENTER THE ELEMENTS OF THE PLANT MATRIX--A.

   A(1,1) =
0

   A(1,2) =
1.

   A(1,3) =
0

   A(1,4) =
0

   A(2,1) =
0
```

```
   A(2,2) =
0

   A(2,3) =
1.

   A(2,4) =
0

   A(3,1) =
0

   A(3,2) =
0

   A(3,3) =
0

   A(3,4) =
1.

   A(4,1) =
0

   A(4,2) =
-15.

   A(4,3) =
-23.

   A(4,4) =
-9.

    THE A  MATRIX (PLANT MATRIX)

0.0          1.00E+00  0.0           0.0
0.0          0.0       1.00E+00      0.0
0.0          0.0       0.0           1.00E+00
0.0        -1.50E+01 -2.30E+01 -9.00E+00

   DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE NUMBER OF COLUMNS OF THE B MATRIX.
1

   ENTER THE ELEMENTS OF THE B MATRIX.
   B(1,1) =
0

   B(2,1) =
0

   B(3,1) =
0

   B(4,1) =
1.

    THE B  MATRIX

 0.0
 0.0
 0.0
 1.00E+00

   DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n
```

```
   ENTER THE NUMBER OF OUTPUTS.
2

   ENTER THE ELEMENTS OF THE C MATRIX.
   C(1,1) =
1.

   C(1,2) =
0

   C(1,3) =
0

   C(1,4) =
0

   C(2,1) =
0

   C(2,2) =
1.

   C(2,3) =
0

   C(2,4) =
0

    THE C  MATRIX
1.00E+00  0.0        0.0        0.0
0.0       1.00E+00   0.0        0.0

   DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    OBSERVABILITY INDEX =   3

    THE SYSTEM (A,B) IS CONTROLLABLE
****************************************************************
   DO YOU  WANT TO RUN OBSCON AGAIN?
n

   THIS CONCLUDES THE OBSERVABILITY INDEX AND
   CONTROLLABILITY PROGRAM (OBSCON)

       COMMENT;  results indicate an observability index of
       3;   this  permits  the design  of  a  second  order
       observer;  eigenvalues  of -3.5 and -4.0  are chosen
       (not  equal to  those  of  the plant)   as  observer
       eigenvalues

   ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
y

LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
   BASIC MATRIX MANIPULATION - <BASMAT>
   RATIONAL TIME RESPONSE - <RTRESP>
   STATE VARIABLE FEEDBACK - <STVAR>
   CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
   LUENBERGER OBSERVER - <LUEN>
   OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
   DISCRETE TIME KALMAN FILTER - <KALMAN>
   OPTIMAL CONTROL - <OPTCON>
   PARTIAL FRACTION EXPANSION - <PRFEXP>
   ROOTS OF A POLYNOMIAL - <ROOTS>
TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
SYMBOLS < >.
luen
```

```
          LUEN IS USED TO DESIGN LUENBERGER OBSERVERS TO
          ACHIEVE A GIVEN CLOSED-LOOP TRANSFER FUNCTION WHEN SOME
          STATE VARIABLES ARE INACCESSIBLE.

          FIRST ENTER THE PROBLEM IDENTIFICATION
          (NOT TO EXCEED 20 CHARACTERS).
thesis example

          ENTER THE ORDER OF THE SYSTEM (UP TO 8).
4

          ENTER THE NUMBER OF MEASUREMENTS (UP TO 8).
2

          ENTER THE ORDER OF THE OBSERVER (UP TO 8).
2

          ENTER THE ELEMENTS OF THE PLANT MATRIX--A.

          A(1,1) =
0

          A(1,2) =
1.

          A(1,3) =
0

          A(1,4) =
0

          A(2,1) =
0

          A(2,2) =
0

          A(2,3) =
1.

          A(2,4) =
0

          A(3,1) =
0

          A(3,2) =
0

          A(3,3) =
0

          A(3,4) =
1.

          A(4,1) =
0

          A(4,2) =
-15.

          A(4,3) =
-23.

          A(4,4) =
-9.
```

```
     THE A  MATRIX (PLANT MATRIX)
            0.0        1.00E+00  0.0         0.0
            0.0        0.0        1.00E+00   0.0
            0.0        0.0        0.0         1.00E+00
            0.0       -1.50E+01 -2.30E+01 -9.00E+00

    DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    ENTER THE ELEMENTS OF THE DISTRIBUTION MATRIX--B.

    B(1)=
0

    B(2)=
0

    B(3)=
0

    B(4)=
1.

     THE B  MATRIX (DISTRIBUTION MATRIX)
            0.0
            0.0
            0.0
            1.00E+00

    DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    ENTER THE ELEMENTS OF THE OUTPUT MATRIX--C.

    C(1,1)=
1.

    C(1,2)=
0

    C(1,3)=
0

    C(1,4)=
0

    C(2,1)=
0

    C(2,2)=
1.

    C(2,3)=
0

    C(2,4)=
0

     THE C  MATRIX (OUTPUT MATRIX)
            1.00E+00   0.0        0.0         0.0
            0.0        1.00E+00  0.0         0.0
    DO YOU  WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    ENTER THE DESIRED  FEEDBACK COEFFICIENTS.

    FDBK COEFF(1)=
20.
```

```
      FDBK COEFF(2)=
13.

      FDBK COEFF(3)=
-6.

      FDBK COEFF(4)=
-3.

       THE DESIRED FEEDBACK COEFFICIENTS

           2.00E+01
           1.30E+01
          -6.00E+00
          -3.00E+00
      DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

      THE OBSERVER EIGENVALUES (F MATRIX) CAN BE SUPPLIED
      EITHER IN THE FORM OF A CHARACTERISTIC POLYNOMIAL OR IN
      THE ROOTS OF THAT POLYNOMIAL.

      ENTER EITHER A P FOR POLYNOMIAL COEFFICIENT FORM
      OR AN F FOR  FACTORED ROOT FORM.
f

      ENTER THE REAL PART OF THE ROOT.
-3.5

      ENTER THE MAGNITUDE OF THE IMAGINARY ROOT.
0

      ENTER THE REAL PART OF THE ROOT.
-4.

      ENTER THE MAGNITUDE OF THE IMAGINARY ROOT.
0

       OBSERVER EIGENVALUES                  REAL PART IMAG. PART
                                             -3.50E+00  0.0
                                             -4.00E+00  0.0

       THE OBSERVER CHARACTERISTIC POLYNOMIAL
       COEFFICIENTS IN ASCENDING POWERS OF S
           1.40E+01  7.50E+00  1.00E+00

*************************************************************

       THE F MATRIX (OBSERVER EIGENVALUE MATRIX)
          -7.50E+00  1.00E+00
          -1.40E+01  0.0

       THE G1 MATRIX (OBSERVER GAIN MATRIX)
           8.55E+01  2.92E+01
           0.0       0.0

       THE G2 MATRIX (OBSERVER GAIN MATRIX)
          -3.00E+00
          -1.50E+00

       OUTPUT FEEDBACK COEFFICIENTS
           2.00E+01  8.50E+00

       COMPENSATOR FEEDBACK COEFFICIENTS
           1.00E+00  0.0

      THIS CONCLUDES THE LUENBERGER OBSERVER DESIGN PROGRAM.
```

```
    DO YOU WANT TO RUN THE PROGRAM AGAIN?
n
    ANALYSIS IS COMPLETE. DO YOU WANT TO RUN LINCON AGAIN?
n
    LINCON IS NOW TERMINATED.
```

The results of STVAR, OBSCON and LUEN are shown in Appendix I. From these the the observer is given as

$$y(t) = \begin{bmatrix} -7.5 & 1.0 \\ -14.0 & 0.0 \end{bmatrix} \begin{bmatrix} y_3(t) \\ y_4(t) \end{bmatrix} + \begin{bmatrix} 85.5 & 29.25 \\ 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} -3.0 \\ -1.5 \end{bmatrix} u(t)$$

and

$$u(t) = 1.0\{r(t) - [20.0 \;\; 8.5] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} - [1.0 \;\; 0.0] \begin{bmatrix} y_3(t) \\ y_4(t) \end{bmatrix}\}$$

G. OPTIMAL CONTROL PROGRAM (OPTCON)

Given the linear, time-invariant system represented as

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{B}u(t) \qquad (4.G-1)$$

OPTCON will minimize the cost function

$$J(N) = 1/2\underline{x}\ (N)\underline{Q}\underline{x}(N) + 1/2\sum_{k=0}^{N-1}[\underline{x}(k)\ \underline{Q}\underline{x}(k) + Ru^2(k)] \qquad (4.G-2)$$

where

$\underline{x}$ = state vector

$\underline{Q}$ = measurement noise covariance matrix (n x n)

N = number of time intervals over which the sum is made

R = random input (a scalar)

$\underline{A}$ = plant matrix (n x n)

$\underline{B}$ = distribution matrix (n x 1)

and

u(t) = control (a scalar).

The output of the program is the feedback gain matrix which, when multiplied by the state vector, yields a scalar control. The following recursive equations were derived using dynamic programming, starting at the terminal time and working backwards.

$$\underline{P}(k) = \underline{\Psi}^T(k)\underline{P}(k-1)\underline{\Psi}(k) + \underline{Q} + R\underline{A}(k)\underline{A}^T(k),\ \underline{P}(0) = 0 \qquad (4.G-3)$$

$$\underline{\psi}(k) = \underline{\Phi} + \underline{\Delta}\underline{A}^T(k), \quad \underline{\psi}(0) = 0 \qquad (4.G-4)$$

$$\underline{A}^T(k) = -[\underline{\Delta}\underline{P}(k-1)\underline{\Phi}]/[\underline{\Delta}^T\underline{P}(k-1)\underline{\Delta} + R], \quad \underline{A}^T(0) = 0 \qquad (4.G-4)$$

For simplicity in programming, the following terms are defined:

$$\text{terminal} = 1/2\underline{x}^T(N)\underline{Q}\underline{x}(N)$$

$$\text{trajectory} = 1/2\sum_{k=0}^{N-1}\underline{x}(k)\ \underline{Q}\underline{x}(k)$$

$$\text{fuel} = 1/2\sum_{k=0}^{N-1}Ru^2(k)$$

1. Terminal Session Example

Given the system and parameters described below find the discrete steady state gains for a sample of 0.1.

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0.0 & 1.0 \\ 1.0 & 0.0 \end{bmatrix}\underline{x}(t) + \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}u(t)$$

$$\underline{Q} = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix}$$

$$R = 1.0$$

In addition, run the program for a time interval of 40.

```
lincon
EXECUTION BEGINS...
```

```
LINCON CONSISTS OF THE FOLLOWING SUBPROGRAMS:
   BASIC MATRIX MANIPULATION - <BASMAT>
   RATIONAL TIME RESPONSE - <RTRESP>
   STATE VARIABLE FEEDBACK - <STVAR>
   CONTROLLABILITY AND OBSERVABILITY - <OBSCON>
   LUENBERGER OBSERVER - <LUEN>
   OPTIMAL CONTROL/KALMAN FILTERS - <RICATI>
   DISCRETE TIME KALMAN FILTER - <KALMAN>
   OPTIMAL CONTROL - <OPTCON>
   PARTIAL FRACTION EXPANSION - <PRFEXP>
   ROOTS OF A POLYNOMIAL - <ROOTS>
TO USE ONE OF THE SUBPROGRAMS ENTER THE NAME BETWEEN THE
SYMBOLS < >.
optcon

   OPTCON MINIMIZES THE THE FOLLOWING COST FUNCTION:

      J(N)=MIN(SUM(X(N)T*Q*X(N)+UT(N-1)*R*U(N-1)))

   THE OUTPUT OF THE PROGRAM IS THE FEEDBACK GAIN MATRIX,
   A TRANSPOSE, WHICH WHEN MULTIPLIED BY THE STATE VECTOR
   YIELDS A SCALAR CONTROL.
   THE FOLLOWING RECURSIVE EQUATIONS WERE DERIVED USING
   DYNAMIC PROGRAMMING, STARTING AT THE TERMINAL TIME AND
   WORKING BACKWARDS:
      (1)  AT(K)=-(DELT*P(K-1)*PHI/(DELT*P(K-1)*DEL+R
                                                AT(0)=0
      (2)     PSI(K)=PHI+DEL*AT(K)                         PSI(0)=
0
      (3)  P(K)=PSIT(K)*P(K-1)*PSI(K)+Q+R*A(K)*AT(K)
                                                P(0)=0

   FIRST ENTER THE PROBLEM IDENTIFICATION
   (*NOT TO EXCEED 20 CHARACTERS*).  thesis example

   ENTER THE NUMBER OF TIME INTERVALS (NSTAGE) OVER WHICH
   THE SUM IS TO BE MADE.
      *** NSTAGE MUST BE ENTERED IN I3 FORMAT ***
      ***(I.E., RIGHT JUSTIFY TO THREE DIGITS ***
 040

   ENTER THE ORDER OF THE SYSTEM (UP TO 8).
      *** ENTER IN I1 FORMAT ***
 2

   ENTER THE ELEMENTS OF THE Q MATRIX.
      *** ALL MATRICES ARE ENTERED IN F-FORMAT ***
      *** I.E., PUT A DECIMAL POINT AFTER YOUR NUMBER ***
   Q(1,1)=
 1.

   Q(1,2)=
 0

   Q(2,1)=
 0

   Q(2,2)=
 2.

    THE Q MATRIX
    1.00E+00  0.0
    0.0       2.00E+00
    DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
 n

   ENTER THE VALUE OF THE SCALAR R
   R =
 1.
```

```
   ENTER THE SAMPLE INTERVAL--DT.
      *** ENTER DT IN F-FORMAT ***
.1

AT THIS POINT YOU MUST CHOOSE ONE OF THE FOLLOWING OPTIONS:
      OPTION A: ENTER THE NUMBER 0 IF
        (1) R IS FINITE, COST=TERMINAL+TRAJECTORY+FUEL,
      OR IF
        (2) R IS ZERO, COST=TERMINAL+TRAJECTORY+ 0

      OPTION B: ENTER THE NUMBER 1 IF
        (1) R IS FINITE, COST=TERMINAL+ 0 + FUEL,
      OR IF
        (2) R IS ZERO, COST=TERMINAL + 0 + 0

   IF YOU WANT TO READ IN THE A AND B MATRICES,
         BUT NOT THE PHI AND DEL MATRICES, ENTER A 0

   HOWEVER, IF YOU WANT TO ENTER THE PHI AND DEL MATRICES,
         BUT NOT THE A AND B MATRICES, ENTER A 1.

   ENTER THE ELEMENTS OF THE PLANT MATRIX--A.
   A(1,1) =
0

   A(1,2) =
1.

   A(2,1) =
1.

   A(2,2) =
0

    THE A  MATRIX (PLANT MATRIX)
   0.0         1.00E+00
   1.00E+00    0.0

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

   ENTER THE ELEMENTS OF THE DISTRIBUTION MATRIX--B.
   B(1,1) =
0

   B(2,1) =
1.

    THE B  MATRIX (DISTRIBUTION MATRIX)
   0.0
   1.00E+00

   DO YOU WANT TO CHANGE ANY ELEMENT OF THE MATRIX?
n

    THE PHI MATRIX
      1.005003E+00      1.001667E-01
      1.001667E-01      1.005003E+00

    THE DEL MATRIX
      5.004164E-03
      1.001667E-01

   MINIMIZATION OVER ALL STAGES
   U(   1)  = AT(J)*X(J)
   AT(1)  =   -2.46E-02
   AT(2)  =   -1.98E-01
```

```
U(  2)  = AT(J)*X(J)
AT(1) =  -7.72E-02
AT(2) =  -3.91E-01

U(  3)  = AT(J)*X(J)
AT(1) =  -1.55E-01
AT(2) =  -5.79E-01

U(  4)  = AT(J)*X(J)
AT(1) =  -2.54E-01
AT(2) =  -7.58E-01

U(  5)  = AT(J)*X(J)
AT(1) =  -3.70E-01
AT(2) =  -9.29E-01

U(  6)  = AT(J)*X(J)
AT(1) =  -4.98E-01
AT(2) =  -1.09E+00

U(  7)  = AT(J)*X(J)
AT(1) =  -6.34E-01
AT(2) =  -1.24E+00

U(  8)  = AT(J)*X(J)
AT(1) =  -7.74E-01
AT(2) =  -1.38E+00

U(  9)  = AT(J)*X(J)
AT(1) =  -9.12E-01
AT(2) =  -1.50E+00

U( 10)  = AT(J)*X(J)
AT(1) =  -1.05E+00
AT(2) =  -1.61E+00

U( 11)  = AT(J)*X(J)
AT(1) =  -1.18E+00
AT(2) =  -1.72E+00

U( 12)  = AT(J)*X(J)
AT(1) =  -1.30E+00
AT(2) =  -1.81E+00

U( 13)  = AT(J)*X(J)
AT(1) =  -1.41E+00
AT(2) =  -1.89E+00

U( 14)  = AT(J)*X(J)
AT(1) =  -1.51E+00
AT(2) =  -1.96E+00

U( 15)  = AT(J)*X(J)
AT(1) =  -1.60E+00
AT(2) =  -2.02E+00

U( 16)  = AT(J)*X(J)
AT(1) =  -1.68E+00
AT(2) =  -2.07E+00

U( 17)  = AT(J)*X(J)
AT(1) =  -1.76E+00
AT(2) =  -2.12E+00

U( 18)  = AT(J)*X(J)
AT(1) =  -1.82E+00
AT(2) =  -2.16E+00
```

```
U( 19) = AT(J)*X(J)
AT(1) =  -1.88E+00
AT(2) =  -2.20E+00

U( 20) = AT(J)*X(J)
AT(1) =  -1.93E+00
AT(2) =  -2.23E+00

U( 21) = AT(J)*X(J)
AT(1) =  -1.97E+00
AT(2) =  -2.25E+00

U( 22) = AT(J)*X(J)
AT(1) =  -2.01E+00
AT(2) =  -2.27E+00

U( 23) = AT(J)*X(J)
AT(1) =  -2.04E+00
AT(2) =  -2.29E+00

U( 24) = AT(J)*X(J)
AT(1) =  -2.07E+00
AT(2) =  -2.31E+00

U( 25) = AT(J)*X(J)
AT(1) =  -2.09E+00
AT(2) =  -2.32E+00

U( 26) = AT(J)*X(J)
AT(1) =  -2.11E+00
AT(2) =  -2.34E+00

U( 27) = AT(J)*X(J)
AT(1) =  -2.13E+00
AT(2) =  -2.35E+00

U( 28) = AT(J)*X(J)
AT(1) =  -2.15E+00
AT(2) =  -2.35E+00

U( 29) = AT(J)*X(J)
AT(1) =  -2.16E+00
AT(2) =  -2.36E+00

U( 30) = AT(J)*X(J)
AT(1) =  -2.17E+00
AT(2) =  -2.37E+00

U( 31) = AT(J)*X(J)
AT(1) =  -2.18E+00
AT(2) =  -2.37E+00

U( 32) = AT(J)*X(J)
AT(1) =  -2.19E+00
AT(2) =  -2.38E+00

U( 33) = AT(J)*X(J)
AT(1) =  -2.20E+00
AT(2) =  -2.38E+00

U( 34) = AT(J)*X(J)
AT(1) =  -2.20E+00
AT(2) =  -2.39E+00

U( 35) = AT(J)*X(J)
AT(1) =  -2.21E+00
AT(2) =  -2.39E+00
```

```
U( 36)  = AT(J)*X(J)
AT(1)  =  -2.21E+00
AT(2)  =  -2.39E+00

U( 37)  = AT(J)*X(J)
AT(1)  =  -2.22E+00
AT(2)  =  -2.39E+00

U( 38)  = AT(J)*X(J)
AT(1)  =  -2.22E+00
AT(2)  =  -2.40E+00

U( 39)  = AT(J)*X(J)
AT(1)  =  -2.22E+00
AT(2)  =  -2.40E+00

U( 40)  = AT(J)*X(J)
AT(1)  =  -2.23E+00
AT(2)  =  -2.40E+00

THIS CONCLUDES THE OPTIMAL CONTROL PROGRAM (OPTCON).
DO YOU WANT TO RUN THE PROGRAM AGAIN?  n

ANALYSIS IS COMPLETE.  DO YOU  WANT TO RUN LINCON AGAIN?
n
LINCON IS NOW TERMINATED.
```

The results, shown in Appendix J, indicate that steady state gains are achieved at about -2.23 and -2.4.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Although LINCON was written primarily as a teaching/learning tool, it can still be quite useful to the practicing engineer for design and analysis problems. It was written in modular form so that it could be easily modified by the addition of subroutines.

LINCON has been extensively tested in an advanced optimal estimation course. The interactive aspects proved highly successful. Hopefully all the "bugs" have been eliminated.

As stated earlier, although the original intent of this thesis was to adapt Desjardins' version of Melsa's LINCON by making it interactive, LINCON began to grow as other routines were added and/or extensively modified.

OPTCON, the optimal control program using recursive equations derived from dynamic programming, is a new member to the LINCON family. Desjardins' KALMAN, the discrete time Kalman filter program, underwent considerable programming changes before achieving its present form.

B. RECOMMENDATIONS

(1) The source program, LINCON FORTRAN A1, must now be passed from user to user and then compiled before it can be used. As can be seen in the terminal session examples, the program was invoked by typing "lincon". Actually, this is an executive program, LINCON EXEC A1, comprising of the following statements:

```
FILEDEF 09 PRINTER (RECFM FA LRECL 133 BLOCK 133

LOAD LINCON (START
```

The first statement defines 09 as the printer and permits it to print out 133 characters per line. The second statement invokes the compiled version of LINCON. It is recommended that LINCON be placed on a utility disk so that users may link to it instead of the current procedure.

(2) At times it can be exceedingly difficult to interpret the tabular output of some of the programs. It is recommended that a graphics package be developed for RTRESP, RICATI, KALMAN and OPTCON. The package should be interactive with the output being first displayed on the terminal screen and then allowing the user to choose the type of output, i.e., VERSATEC, TEKTRONIX or print-plot.

(3) In general, the programs are limited to eighth-order problems. If the need should arise to solve higher order systems, this limitation may be removed by extending the appropriate dimension statements. The user must remember to alter the format statement pertaining to the output, either decreasing the significant figures or adding a "wrap-around" feature to overcome the printer limitations of 133 characters per line.

(4) As it is written, LUEN can only solve for a reduced-order observer. It is recommended that the program be modified so that the user has the option of selecting a reduced-order observer or an identity observer.

(5) The ficticious and real c matrices of SIVAR are required to have the dimensions 1 x n. It is recommended that the program be modified to accept a dimension size of m x n.

APPENDIX A

```
     BASIC MATRIX PROGRAM
     PROBLEM IDENTIFICATION:     THESIS EXAMPLE
**********************************************************
     THE A MATRIX
       1.000000E+00    0.0             0.0
       0.0            -2.000000E+00   1.000000E+00
       0.0            -5.000000E-01   1.000000E+00

     THE DETERMINANT OF THE MATRIX
      -1.500000E+00

     THE INVERSE OF THE MATRIX
       1.000000E+00    0.0             0.0
       0.0            -6.666665E-01   6.666665E-01
       0.0            -3.333333E-01   1.333333E+00

**********************************************************
     THE MATRIX COEFFICIENTS OF THE NUMERATOR OF THE
     PHI(S) MATRIX
     THE MATRIX COEFFICIENT OF S**2
       1.000000E+00    0.0             0.0
       0.0             1.000000E+00   0.0
       0.0             0.0             1.000000E+00

     THE MATRIX COEFFICIENT OF S**1
       1.000000E+00    0.0             0.0
       0.0            -2.000000E+00   1.000000E+00
       0.0            -5.000000E-01   1.000000E+00

     THE MATRIX COEFFICIENT OF S**0
      -1.500000E+00    0.0             0.0
       0.0             1.000000E+00  -1.000000E+00
       0.0             5.000000E-01  -2.000000E+00

**********************************************************
     THE CHARACTERISTIC POLYNOMIAL-IN ASCENDING POWERS OF S
       1.500000E+00   -2.500000E+00   0.0      1.000000E+00
***********************************************************
     THE EIGENVALUES OF THE A MATRIX
        REAL PART     IMAGINARY PART
       8.228755E-01    0.0
      -1.822876E+00    0.0
       1.000000E+00    0.0
**********************************************************
     THE ELEMENTS OF THE STATE TRANSITION MATRIX
THE MATRIX COEFFICENT OF EXP(    8.228755E-01)T
       0.0             0.0             0.0
       0.0            -6.694317E-02   3.779633E-01
       0.0            -1.889811E-01   1.066945E+00

THE MATRIX COEFFICENT OF EXP(   -1.822876E+00)T
       0.0             0.0             0.0
       0.0             1.066944E+00  -3.779641E-01
       0.0             1.889820E-01  -6.694669E-02

THE MATRIX COEFFICENT OF EXP(    1.000000E+00)T
       1.000000E+00    0.0             0.0
       0.0            -2.264977E-06   2.861023E-06
       0.0            -5.364418E-07   1.072884E-06
```

APPENDIX B

```
     PARTIAL FRACTION EXPANSION
     PROBLEM IDENTIFICATION:     THESIS EXAMPLE
**********************************************************
     THE INPUT FUNCTION GAIN
       7.000000E+00
**********************************************************
     NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
   2.0000E+00    2.0000E+00    1.0000E+00    1.0000E+00

     NUMERATOR ROOTS ARE
       REAL PART       IMAG. PART
       0.0            -1.414213E+00
       0.0             1.414213E+00
      -1.000000E+00    0.0
**********************************************************
     THE INPUT FUNCTION GAIN
       7.000000E+00
**********************************************************
     DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
   2.000E+00   1.000E+00     0.0    3.000E+00     1.000E+00

     DENOMINATOR ROOTS ARE
      REAL PART       IMAG. PART    MULTIPLICITY
      -3.018859E+00   0.0                1
      -9.127604E-01   0.0                1
       4.658098E-01   -8.308427E-01      1
       4.658098E-01    8.308427E-01      1
**********************************************************
     RESIDUE MATRIX - REAL PART
       5.810872E+00
       3.170837E-01
       4.360231E-01
       4.360231E-01

     RESIDUE MATRIX - IMAG. PART
      -2.050349E-07
       0.0
       2.061406E+00
      -2.061406E+00
**********************************************************
```

APPENDIX C

```
      ROOTS OF A POLYNOMIAL
      PROBLEM IDENTIFICATION:      THESIS EXAMPLE
***********************************************************************
      POLYNOMIAL COEFFICIENTS - IN ASCENDING POWERS OF S
        8.0000E-01     5.1000E+00     3.0000E+00     1.0000E+00

      THE ROOTS ARE             REAL PART          IMAGINARY PART
                               -1.413221E+00      1.616248E+00
                               -1.413221E+00     -1.616248E+00
                               -1.735564E-01      0.0
***********************************************************************
```

APPENDIX D

```
     RATIONAL TIME RESPONSE
     PROBLEM IDENTIFICATION:      THESIS EXAMPLE
***********************************************************
     THE A MATRIX (PLANT MATRIX)
       1.000000E+00     0.0                 0.0
       0.0             -2.000000E+00        1.000000E+00
       0.0             -5.000000E-01        1.000000E+00

     THE B MATRIX (CONTROL VECTOR)
       0.0
       0.0
       1.000000E+00

     THE C MATRIX (OUTPUT VECTOR)
       1.000000E+00
       4.000000E-01
       1.000000E+00

     THE FEEDBACK COEFFICIENT VECTOR
       1.000000E+00
       1.000000E+00
       0.0

     THE CONTROLLER GAIN
       3.200000E+00
***********************************************************
     INITIAL CONDITIONS VECTOR-X(0)
       0.0
       0.0
       0.0

     THE INPUT FUNCTION GAIN
       5.000000E-01

     NUMERATOR POLYNOMIAL OF R(S)-ASCENDING POWERS OF S
       1.000000E+00      1.000000E+00

     NUMERATOR ROOTS ARE
       REAL PART         IMAG. PART
      -1.000000E+00      0.0

     DENOMINATOR POLYNOMIAL OF R(S) - ASCENDING POWERS OF S
      -2.000000E+00      0.0                1.000000E+00

     DENOMINATOR ROOTS ARE
       REAL PART         IMAG. PART
       1.414213E+00      0.0
      -1.414213E+00      0.0
***********************************************************
     THE TIME RESPONSE OF THE STATE X(T)
     VECTOR COEFFICIENT OF EXP(-5.00E-01)T*COS(1.20E+00)T
       0.0             -3.695463E-01      -9.717697E-01

     VECTOR COEFFICIENT OF EXP(-5.00E-01)T*SIN(1.20E+00)T
       0.0             -3.466749E-01      -7.502007E-02

     THE VECTOR COEFFICIENT OF EXP(      1.000000E+00)T
       0.0               0.0                2.682209E-06

     THE VECTOR COEFFICIENT OF EXP(      1.414213E+00)T
       0.0               2.670370E-01       9.117225E-01

     THE VECTOR COEFFICIENT OF EXP(     -1.414213E+00)T
       0.0               1.025083E-01       6.004864E-02
***********************************************************
```

```
     THE TIME RESPONSE OF THE OUTPUT Y(T)
     THE COEFFICIENT OF EXP(-5.0000E-01)T*COS(1.2041E+00)T
      -1.119588E+00

     THE COEFFICIENT OF EXP(-5.0000E-01)T*SIN(1.2041E+00)T
      -2.136900E-01

     THE COEFFICIENT OF EXP(    1.000000E+00)T
       2.682209E-06

     THE COEFFICIENT OF EXP(    1.414213E+00)T
       1.018537E+00

     THE COEFFICIENT OF EXP(   -1.414213E+00)T
       1.010521E-01
********************************************************
```

APPENDIX E

```
     OBSERVABILITY, CONTROLLABILITY
     PROBLEM IDENTIFICATION:     THESIS EXAMPLE
*****************************************************************
     THE A MATRIX (PLANT MATRIX)
  0.0               1.000000E+00     0.0
 -1.000000E+00     -5.000000E-01     1.000000E+00
  0.0               0.0              1.000000E+00

     THE B MATRIX
  2.000000E+00      1.000000E+00
  0.0               1.000000E+00
  0.0               0.0

     THE C MATRIX
  0.0               1.000000E+00     0.0
  1.000000E+00      1.000000E+00     0.0
 -2.000000E+00      1.000000E+00     0.0
*****************************************************************
     OBSERVABILITY INDEX =   2
     THE SYSTEM (A,B) IS UNCONTROLLABLE
*****************************************************************
```

APPENDIX F

```
    OPTIMAL CONTROL/CONTINUOUS KALMAN FILTER PROGRAM
    PROBLEM IDENTIFICATION:     THESIS EXAMPLE
*************************************************************************
    THE A MATRIX (PLANT MATRIX)
 -1.000000E+00     0.0
  0.0             -2.000000E+00

    THE B MATRIX (DISTRIBUTION MATRIX)
  1.000000E+00     0.0
  0.0              1.000000E+00

    THE C MATRIX (MEASUREMENT MATRIX)
  1.000000E+00     0.0
  0.0              2.000000E+00
*************************************************************************
    THE R MATRIX (CONTROL WEIGHTING MATRIX)
  1.000000E+00     0.0
  0.0              2.000000E+00

    THE Q MATRIX (STATE WEIGHTING MATRIX)
  1.000000E+00     1.000000E+00
  1.000000E+00     1.000000E+00
   ***********************
   *** CONTROL OPTION ***
   ***********************
    THE P MATRIX (TERMINAL BOUNDARY VALUE MATRIX)
  0.0              0.0
  0.0              0.0
*************************************************************************
    TRANSIENT SOLUTION
   TIME =  1.000E+01
    GAINS
 0.0          0.0
 0.0          0.0

   TIME =  9.000E+00
    GAINS
  3.745150E-01     2.808074E-01
  1.404037E-01     1.112729E-01

   TIME =  8.000E+00
    GAINS
  3.982797E-01     2.849711E-01
  1.424856E-01     1.117260E-01

   TIME =  7.000E+00
    GAINS
  3.997058E-01     2.847993E-01
  1.423996E-01     1.117371E-01

   TIME =  6.000E+00
    GAINS
  3.997959E-01     2.847759E-01
  1.423879E-01     1.117399E-01

   TIME =  5.000E+00
    GAINS
  3.997999E-01     2.847756E-01
  1.423877E-01     1.117399E-01


   TIME =  4.000E+00
    GAINS
  3.997999E-01     2.847756E-01
  1.423877E-01     1.117399E-01
```

```
    TIME =  3.000E+00
     GAINS
   3.997999E-01     2.847756E-01
   1.423877E-01     1.117399E-01

    TIME =  2.000E+00
     GAINS
   3.997999E-01     2.847756E-01
   1.423877E-01     1.117399E-01

    TIME =  1.000E+00
     GAINS
   3.997999E-01     2.847756E-01
   1.423877E-01     1.117399E-01

    TIME =  2.861E-06
     GAINS
   3.997999E-01     2.847756E-01
   1.423877E-01     1.117399E-01
*********************************************************
    *********************
    *** FILTER OPTION ***
    *********************
     THE P MATRIX (INITIAL BOUNDARY VALUE MATRIX)
   0.0              0.0
   0.0              0.0
*********************************************************
     TRANSIENT SOLUTION
    TIME =  0.0
     GAINS
 0.0       0.0
 0.0       0.0

    TIME =  5.000E-01
     GAINS
   2.393162E-01     2.004539E-01
   8.018157E-01     6.849594E-01

    TIME =  1.000E+00
     GAINS
   2.708317E-01     2.127941E-01
   8.511767E-01     7.053521E-01

    TIME =  1.500E+00
     GAINS
   2.768488E-01     2.133181E-01
   8.532724E-01     7.055743E-01

    TIME =  2.000E+00
     GAINS
   2.784715E-01     2.132589E-01
   8.530354E-01     7.055843E-01

    TIME =  2.500E+00
     GAINS
   2.789688E-01     2.132211E-01
   8.528843E-01     7.055938E-01

    TIME =  3.000E+00
     GAINS
   2.791271E-01     2.132074E-01
   8.528297E-01     7.055972E-01


    TIME =  3.500E+00
     GAINS
   2.791781E-01     2.132027E-01
   8.528109E-01     7.055984E-01
```

```
    TIME =  4.000E+00
     GAINS
   2.791938E-01    2.132020E-01
   8.528078E-01    7.055986E-01

    TIME =  4.500E+00
     GAINS
   2.791983E-01    2.132018E-01
   8.528070E-01    7.055986E-01

    TIME =  5.000E+00
     GAINS
   2.791983E-01    2.132018E-01
   8.528070E-01    7.055986E-01
***********************************************************
```

APPENDIX G

```
    DISCRETE TIME KALMAN FILTER PROGRAM
    PROBLEM IDENTIFICATION:      THESIS EXAMPLE
*********************************************************************
    THE PHI MATRIX (TRANSITION MATRIX)
 1.000000E+00     5.000000E-01
 0.0              1.000000E+00

    THE GAMMA MATRIX (DISTRIBUTION MATRIX)
 1.250000E-01
 5.000000E-01

    THE W MATRIX (MEAN-SQUARE MAGNITUDE OF THE
    PERTURBATION ACCELERATION MATRIX)
 4.000000E+00

    THE H MATRIX (OBSERVATION MATRIX)
 1.000000E+00     0.0
*********************************************************************
    THE R MATRIX (MEASUREMENT NOISE COVARIANCE MATRIX)
 5.000000E+00

    THE P MATRIX (INITIAL CONDITION MATRIX)
 1.000000E+03     0.0
 0.0              1.000000E+03
*********************************************************************
     K =    0
    GAINS
 9.950248E-01     0.0

     K =    1
    GAINS
 9.807723E-01     1.923759E+00

     K =    2
    GAINS
 8.290299E-01     9.980751E-01

     K =    3
    GAINS
 7.028207E-01     6.237227E-01

     K =    4
    GAINS
 6.134719E-01     4.522047E-01

     K =    5
    GAINS
 5.543699E-01     3.716016E-01

     K =    6
    GAINS
 5.184991E-01     3.362513E-01

     K =    7
    GAINS
 4.992278E-01     3.232893E-01

     K =    8
    GAINS
 4.904468E-01     3.201340E-01


     K =    9
    GAINS
 4.872549E-01     3.202326E-01
```

```
   K =  10
  GAINS
4.864400E-01    3.207918E-01

   K =  11
  GAINS
4.863475E-01    3.210392E-01

   K =  12
  GAINS
4.863456E-01    3.209999E-01

   K =  13
  GAINS
4.862944E-01    3.208563E-01

   K =  14
  GAINS
4.862168E-01    3.207275E-01

   K =  15
  GAINS
4.861480E-01    3.206494E-01

   K =  16
  GAINS
4.861035E-01    3.206143E-01

   K =  17
  GAINS
4.860806E-01    3.206035E-01

   K =  18
  GAINS
4.860712E-01    3.206024E-01

   K =  19
  GAINS
4.860685E-01    3.206034E-01

   K =  20
  GAINS
4.860680E-01    3.206041E-01
***********************************************************
```

APPENDIX H

```
     STATE VARIABLE FEEDBACK PROGRAM
     PROBLEM IDENTIFICATION:     THESIS EXAMPLE
**********************************************************************
     THE A MATRIX (PLANT MATRIX)
      -1.000000E+00      1.000000E+00      0.0
       0.0               0.0               1.000000E+00
       0.0              -3.000000E+00      0.0

     THE B MATRIX (CONTROL MATRIX)
       0.0
       0.0
       1.000000E+00
**********************************************************************
     OPEN-LOOP CALCULATIONS
     DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
       3.0000E+00      3.0000E+00      1.0000E+00      1.0000E+00

     THE ROOTS ARE                REAL PART          IMAGINARY PART
                                   0.0               -1.732051E+00
                                   0.0                1.732051E+00
                                  -1.000000E+00       0.0
*********************************************************************
     THE C MATRIX (FICTICIOUS OUTPUT VECTOR)
       0.0
       0.0
       1.000000E+00

     NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
       0.0               1.000000E+00      1.000000E+00

     THE ROOTS ARE                REAL PART          IMAGINARY PART
                                  -1.000000E+00       0.0
                                   0.0                0.0
**********************************************************************
     THE C MATRIX (OUTPUT VECTOR)
       1.000000E+00
       1.000000E+00
       0.0

     NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
       2.000000E+00      1.000000E+00

     THE ROOTS ARE                REAL PART          IMAGINARY PART
                                  -2.000000E+00       0.0
**********************************************************************

     CLOSED-LOOP CALCULATIONS
     KEY = P     *****
     THE NUMERATOR OF H-EQUIVALENT -
     IN ASCENDING POWERS OF S
       5.000000E-01      1.500000E+00      1.500000E+00

     THE ROOTS ARE             REAL PART IMAG. PART
                                  -5.000000E-01     -2.886756E-01
                                  -5.000000E-01      2.886756E-01

     THE FEEDBACK COEFFICIENTS
       5.000000E-01      0.0               1.500000E+00

     THE GAIN =      2.000000E+00


     THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL -
     IN ASCENDING POWERS OF S
       4.0000E+00      6.0000E+00      4.0000E+00      1.0000E+00
```

```
THE ROOTS ARE         REAL PART IMAG. PART
                         -2.000000E+00     0.0
                         -1.000000E+00   -1.000000E+00
                         -1.000000E+00    1.000000E+00

MAXIMUM NORMALIZED ERROR =  0.0
```

APPENDIX I

```
    STATE VARIABLE FEEDBACK PROGRAM
    PROBLEM IDENTIFICATION:     THESIS EXAMPLE
************************************************************
    THE A MATRIX (PLANT MATRIX)
      0.0     1.000000E+00     0.0              0.0
      0.0     0.0              1.000000E+00     0.0
      0.0     0.0              0.0              1.000000E+00
      0.0    -1.500000E+01    -2.300000E+01    -9.000000E+00

    THE B MATRIX (CONTROL MATRIX)
      0.0
      0.0
      0.0
      1.000000E+00
************************************************************
    OPEN-LOOP CALCULATIONS
    DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
      0.0    1.500E+01  2.300E+01  9.000E+00   1.000E+00

    THE ROOTS ARE               REAL PART        IMAGINARY PART
                               -3.000000E+00     0.0
                               -4.999999E+00     0.0
                               -9.999999E-01     0.0
                                0.0              0.0
*************************************************************
    THE C MATRIX (OUTPUT VECTOR)
      2.000000E+01
      1.000000E+01
      0.0
      0.0

    NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
      2.000000E+01     1.000000E+01

    THE ROOTS ARE               REAL PART        IMAGINARY PART
                               -2.000000E+00     0.0
***********************************************************
    CLOSED-LOOP CALCULATIONS
    KEY = P    *****
    THE NUMERATOR OF H-EQUIVALENT-IN ASCENDING POWERS OF S
      2.000E+01     1.300E+01    -6.000E+00    -3.000E+00

    THE ROOTS ARE            REAL PART IMAG. PART
                               -1.039623E+00     0.0
                                3.206289E+00     0.0

    THE FEEDBACK COEFFICIENTS
      2.000E+01     1.300E+01    -6.000E+00    -3.000E+00

    THE GAIN =     1.000000E+00

    THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL
              IN ASCENDING POWERS OF S
  2.000E+01   2.800E+01   1.700E+01   6.000E+00   1.000E+00

    THE ROOTS ARE            REAL PART IMAG. PART
                               -1.000000E+00    -2.000000E+00
                               -1.000000E+00     2.000000E+00
                               -2.000484E+00     0.0
                               -1.999516E+00     0.0
    MAXIMUM NORMALIZED ERROR =  0.0


    OBSERVABILITY, CONTROLLABILITY
    PROBLEM IDENTIFICATION:     THESIS EXAMPLE
```

```
*************************************************************************
    THE A MATRIX (PLANT MATRIX)
 0.0          1.000000E+00       0.0               0.0
 0.0          0.0                1.000000E+00      0.0
 0.0          0.0                0.0               1.000000E+00
 0.0         -1.500000E+01      -2.300000E+01     -9.000000E+00

    THE B MATRIX
 0.0
 0.0
 0.0
 1.000000E+00

    THE C MATRIX
 1.000000E+00     0.0                0.0                0.0
 0.0              1.000000E+00       0.0                0.0
*************************************************************************
    OBSERVABILITY INDEX =  3
    THE SYSTEM (A,B) IS CONTROLLABLE
*************************************************************************




    LUENBERGER OBSERVER DESIGN PROGRAM
    PROBLEM IDENTIFICATION:     THESIS EXAMPLE
*************************************************************************
    THE A MATRIX (PLANT MATRIX)
      0.0      1.000000E+00       0.0               0.0
      0.0      0.0                1.000000E+00      0.0
      0.0      0.0                0.0               1.000000E+00
      0.0     -1.500000E+01      -2.300000E+01     -9.000000E+00

    THE B MATRIX (DISTRIBUTION MATRIX)
      0.0
      0.0
      0.0
      1.000000E+00

    THE C MATRIX (OUTPUT MATRIX)
      1.000000E+00     0.0                0.0                0.0
      0.0              1.000000E+00       0.0                0.0

    THE DESIRED FEEDBACK COEFFICIENTS
      2.000000E+01
      1.300000E+01
     -6.000000E+00
     -3.000000E+00

    OBSERVER EIGENVALUES           REAL PART          IMAG. PART
                                   -3.500000E+00       0.0
                                   -4.000000E+00       0.0

    OBSERVER CHARACTERISTIC POLYNOMIAL
    COEFFICIENTS IN ASCENDING POWERS OF S
      1.400000E+01     7.500000E+00     1.000000E+00

    THE F MATRIX (OBSERVER EIGENVALUE MATRIX)
     -7.500000E+00     1.000000E+00
     -1.400000E+01     0.0

    THE G1 MATRIX (OBSERVER GAIN MATRIX)
      8.549997E+01     2.924994E+01
      0.0              0.0

    THE G2 MATRIX (OBSERVER GAIN MATRIX)
     -2.999997E+00
     -1.500001E+00
```

```
OUTPUT FEEDBACK  COEFFICIENTS
   2.000000E+01     8.500001E+00

COMPENSATOR FEEDBACK  COEFFICIENTS
   1.000000E+00     0.0
```

APPENDIX J

```
     OPTIMAL CONTROL PROGRAM
     PROBLEM IDENTIFICATION:     THESIS EXAMPLE
***************************************************************
     THE NUMBER OF TIME INTERVALS =   40

     THE ORDER OF THE SYSTEM = 2

     THE Q MATRIX
       1.000000E+00     0.0
       0.0              2.000000E+00

     THE SCALAR R =   1.

     THE SAMPLE INTERVAL (DT) =      0.1000

     THE A MATRIX (PLANT MATRIX)
       0.0              1.000000E+00
       1.000000E+00     0.0

     THE B MATRIX (DISTRIBUTION MATRIX)
       0.0
       1.000000E+00

     THE PHI MATRIX
       1.005003E+00     1.001667E-01
       1.001667E-01     1.005003E+00

     THE DEL MATRIX
       5.004164E-03
       1.001667E-01

    MINIMIZATION OVER ALL STAGES

    U(   1) = AT(J)*X(J)
    AT(1)  =  -2.46E-02
    AT(2)  =  -1.98E-01

    U(   2) = AT(J)*X(J)
    AT(1)  =  -7.72E-02
    AT(2)  =  -3.91E-01

    U(   3) = AT(J)*X(J)
    AT(1)  =  -1.55E-01
    AT(2)  =  -5.79E-01

    U(   4) = AT(J)*X(J)
    AT(1)  =  -2.54E-01
    AT(2)  =  -7.58E-01

    U(   5) = AT(J)*X(J)
    AT(1)  =  -3.70E-01
    AT(2)  =  -9.29E-01

    U(   6) = AT(J)*X(J)
    AT(1)  =  -4.98E-01
    AT(2)  =  -1.09E+00

    U(   7) = AT(J)*X(J)
    AT(1)  =  -6.34E-01
    AT(2)  =  -1.24E+00

    U(   8) = AT(J)*X(J)
    AT(1)  =  -7.74E-01
    AT(2)  =  -1.38E+00
```

```
U(  9) =  AT(J)*X(J)
AT(1)  =  -9.12E-01
AT(2)  =  -1.50E+00

U( 10) =  AT(J)*X(J)
AT(1)  =  -1.05E+00
AT(2)  =  -1.61E+00

U( 11) =  AT(J)*X(J)
AT(1)  =  -1.18E+00
AT(2)  =  -1.72E+00

U( 12) =  AT(J)*X(J)
AT(1)  =  -1.30E+00
AT(2)  =  -1.81E+00

U( 13) =  AT(J)*X(J)
AT(1)  =  -1.41E+00
AT(2)  =  -1.89E+00

U( 14) =  AT(J)*X(J)
AT(1)  =  -1.51E+00
AT(2)  =  -1.96E+00

U( 15) =  AT(J)*X(J)
AT(1)  =  -1.60E+00
AT(2)  =  -2.02E+00

U( 16) =  AT(J)*X(J)
AT(1)  =  -1.68E+00
AT(2)  =  -2.07E+00

U( 17) =  AT(J)*X(J)
AT(1)  =  -1.75E+00
AT(2)  =  -2.12E+00

U( 18) =  AT(J)*X(J)
AT(1)  =  -1.82E+00
AT(2)  =  -2.16E+00

U( 19) =  AT(J)*X(J)
AT(1)  =  -1.88E+00
AT(2)  =  -2.20E+00

U( 20) =  AT(J)*X(J)
AT(1)  =  -1.93E+00
AT(2)  =  -2.23E+00

U( 21) =  AT(J)*X(J)
AT(1)  =  -1.97E+00
AT(2)  =  -2.25E+00

U( 22) =  AT(J)*X(J)
AT(1)  =  -2.01E+00
AT(2)  =  -2.27E+00

U( 23) =  AT(J)*X(J)
AT(1)  =  -2.04E+00
AT(2)  =  -2.29E+00

U( 24) =  AT(J)*X(J)
AT(1)  =  -2.07E+00
AT(2)  =  -2.31E+00

U( 25) =  AT(J)*X(J)
AT(1)  =  -2.09E+00
AT(2)  =  -2.32E+00
```

```
U( 26) = AT(J)*X(J)
AT(1)  =  -2.11E+00
AT(2)  =  -2.34E+00

U( 27) = AT(J)*X(J)
AT(1)  =  -2.13E+00
AT(2)  =  -2.35E+00

U( 28) = AT(J)*X(J)
AT(1)  =  -2.15E+00
AT(2)  =  -2.35E+00

U( 29) = AT(J)*X(J)
AT(1)  =  -2.16E+00
AT(2)  =  -2.36E+00

U( 30) = AT(J)*X(J)
AT(1)  =  -2.17E+00
AT(2)  =  -2.37E+00

U( 31) = AT(J)*X(J)
AT(1)  =  -2.18E+00
AT(2)  =  -2.37E+00

U( 32) = AT(J)*X(J)
AT(1)  =  -2.19E+00
AT(2)  =  -2.38E+00

U( 33) = AT(J)*X(J)
AT(1)  =  -2.20E+00
AT(2)  =  -2.38E+00

U( 34) = AT(J)*X(J)
AT(1)  =  -2.20E+00
AT(2)  =  -2.39E+00

U( 35) = AT(J)*X(J)
AT(1)  =  -2.21E+00
AT(2)  =  -2.39E+00

U( 36) = AT(J)*X(J)
AT(1)  =  -2.21E+00
AT(2)  =  -2.39E+00

U( 37) = AT(J)*X(J)
AT(1)  =  -2.22E+00
AT(2)  =  -2.39E+00

U( 38) = AT(J)*X(J)
AT(1)  =  -2.22E+00
AT(2)  =  -2.40E+00

U( 39) = AT(J)*X(J)
AT(1)  =  -2.22E+00
AT(2)  =  -2.40E+00

U( 40) = AT(J)*X(J)
AT(1)  =  -2.23E+00
AT(2)  =  -2.40E+00
*********************************************************
```

LIST OF REFERENCES

1. Melsa, James L., and Jones, Stephen K., Computer Programs for Computational Assistance in the Study of Linear Control Theory, McGraw-Hill, 1973.

2. Desjardins, Berthier, A User's Manual for Linear Control Programs on IBM/360, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1979.

3. Goos, G., and Hartmanis, J. Lecture Notes in Computer Science, Interactive Systems, Proceedings, 6th Informatik Symposium, IBM Germany, Bad Homburg v.d.H., September, 1976.

4. Cadzow, J.A. and Martens, H.R., Discrete-Time and Computer Control Systems, Prentice-Hall, 1970.

5. Schultz, D.G., and Melsa, J.L., State Functions and Linear Control Systems, McGraw-Hill, New York, N.Y., 1967.

6. Sage, Andrew P., Optimal System Control, Prentice-Hall, 1968.

7. Melsa, J. L., "An Algorithm for the Analysis and Design of Linear State Variable Feedback Systems", Proc. of Asilomar Conference on Circuits and Systems, Monterey, California, pp. 791 - 799, Nov. 1967.

8. Luenberger, D. G., "Observing the State of a Linear System", IEEE Transactions on Military Electronics, vol. MIL-8, pp. 74-80, April 1964.

9. Kirk, D.E., Optimal Estimation: An Introduction to the Theory and Applications, 1975.

BIBLIOGRAPHY

Hamming, R. W., *Numerical Methods for Scientists and Engineers*, McGraw-Hill Book Co., Inc., 1962.

Huelsman, L. P., *Digital Computations in Basic Circuit Theory*, McGraw-Hill Book Co., Inc., 1968.

Kalman, R. E., "Mathematical Description of Linear Dynamical Systems", *J.S.I.A.M. Control*, Ser. A, Vol. 1, no. 2, pp 152-192, 1963.

Mendel, Jerry M., *Discrete Techniques of Parameter Estimation, The Equation Error Formulation*, Marcel Dekker, Inc., New York, 1973.

Jazwinski, Andrew H., *Stochastic Processes and Filtering Theory*, Academic Press, Inc., New York, 1970.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library Code 0142 Naval Postgraduate School Monterey, Ca 93940	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor Donald E. Kirk, Code 62Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Professor Hal A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, Ca 93940	2
6.	Professor Alex Gerba, Jr., Code 62Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	LT S. Constandoulakis, HN Antoniou Paraskeva 5 Marousi Athens Greece	1
8.	LT Raul E. Samaniego, Ecuadorean Navy P.O. Box 953 Guayaquil, Ecuador South America	1
9.	LTJG Ilker Bayraktar, TN Bayraktar Muhasebe Burosu Iscanlar Carsisi No. 19 Corlu/Tekirdag/Turkey	1
10.	LT Robert M. Thompson, USN Naval Electronic Systems Security Engineering Center 3801 Nebraska Ave. Washington, D.C. 20390	2

EXEC D1

LINCON TEXT 21

200165

Th
T4
c.

200165

Thesis
T4365
c.1

Thompson
A user's manual for interactive linear control programs on IBM/3033.

DEC 3 85
33249

200165

Thesis
T4365
c.1

Thompson
A user's manual for interactive linear control programs on IBM/3033.

A user's manual for interactive linear c
3 2768 001 01105 9
DUDLEY KNOX LIBRARY